Capturing the Past
Inspiring the Future

Pete Lomas

Interviewed by

Richard Sharpe

17th October 2019

At the

WCIT Hall,
32a Bartholomew Close, London, EC1A 7JN
Kindly provided by The Worshipful Company of Information Technologists

Copyright
Archives of IT
(Registered Charity 1164198)

Welcome to the Archives of Information Technology. It is the 17th of October 2019, and we are in central London, in the Livery Hall of the Worshipful Company of Information Technologists. The Archives claim to, and so far have been successful in, capturing the past and inspiring the future, and we'll come back to that inspiration of the future in a moment. I'm Richard Sharpe, and I have been covering and analysing and researching information technology since the 1970s.

[00:39]

Making his contribution to the Archives today is a man who has contributed probably more than anybody else in the country to attempt to inspire the future. He is absolutely dedicated and passionate about getting young people to tinker with hardware and software, and not just to use computers, IT, information technology, et cetera, for their own needs, but also to think of themselves as builders of the technology themselves. And if we're going to have the successful Network of Things, then there's millions of things which are going to need to be connected, and his work as an electrical engineer in his company Norcott Technologies is also involved in that. So we introduce today to the Archives Peter Lomas. Peter, where were you born, and what were your parents doing?

[01:35]

Oh good morning. I was born in Salford, which most people will recognise as being next to Manchester in the North West. My parents were, my mother was a housewife, because I have an older brother, and she was already looking after him when I was born, and my father was a foreman electrician for a building company, George Wimpey. And, I think, the recollection that I have of most of that is, our garage filled with bits of wire, motors, switches, and all the sort of things that he needed, because he was a service engineer as well, so he tended to use our garage as a place to store all those bits, rather than having to go back to the depot every day. The other thing I remember is that he worked in credibly hard. And I really didn't see a lot of him, but when, when I did, the thing was, he was passionate about building things. He built an extension for the house, he helped people wire their houses. He was always a doer. He also tended to the garden. Now gardening's something he never passed down to me, that's not something I really… I think, plants just grow too slowly for me. I like to move on.

But those other things, that passion to build, and particularly build things that have moving parts to them, is part of it?

That's right. I mean, building, building things I think is the thing that defines me, and I think I definitely got that from my father.

From your mother, what did you get?

Oh my mother, just nurturing, and encouragement. They never actually said to me, I don't ever remember them saying, 'You can't do that.' And that's something I've tried to do with my child as well. It's not saying, 'You can't do it.' 'Here's a couple of reasons why perhaps you shouldn't, but, no, there's nothing you can't do.' And I also, you know, I've also brought forward that technically, you know, perhaps nothing is impossible. We just need signs to figure out how it works, and then engineering can actually make some use of it for the benefit of society.
[03:47]
But I think, the thing that probably set things in motion is, at thirteen I was allowed to go into Manchester. Now, in those days Manchester was the centre of electronics and computers. We had ICL, Ferranti, and all the offshoot material that they didn't want ended up in junk shops on a road called Tib Street, and I remember it quite fondly. I would go on a Saturday and I would root. And one day I came home with a great big chunk of a piece of telephone exchange. And the family were absolutely horrified, a) because it was large, and b) I had spent, probably too much money on it. But I had an absolute plan with it. And, I took it to pieces, I cleaned all the parts up, and then somehow I managed to put that together to make a noughts and crosses machine. So you could dial in with a normal telephone dial, it would make the move, and then it would make its counter move. I suppose I didn't realise at the time, but I had just made a fixed-program computer. But this was well before really computers were in the curriculum. I certainly didn't know, you know, see them at school, it was…

You were thirteen then?

Yeah, I was thirteen.

So, this was 1968?

Yes.

Right.

So, I mean you know, we were well…

This, this part of the exchange was all relays was it?

It was relays, uniselectors, diodes, and capacitors. Now the biggest challenge I had was to make a oscillator. But eventually I figured out that a couple of diodes and a capacitor and a relay could be made into an oscillator.

Right.

Just because of the characteristics of a, of a relay, that it, it comes in at a voltage, but lets go at a much lower voltage. And, I think the problem is, the distance in time between that now, I've been desperately trying to recreate it. But the, you know, the inventiveness you have as a child seems to have dissipated a little bit over the years. [laughs] So I've got some of the bits, but I'm now scratching my head how we're going to put that back together.

Right.

But it's a job, it's, it's something I've got to do at some point.

[06:03]
According to my notes, you didn't really like school very much.

[hesitates] No. School for me was, a little bit of a disaster. I was a daydreamer in their eyes, because I was thinking about all sorts of little projects I could make. I mean prior to the electronics, I had been making a thing called marble rooms, where you just take a cardboard box and you bend bits of cardboard, and you make the

marbles run up and down them. That was great for learning the laws of physics, as errant marbles would disappear under furniture or between floorboards. But again, it was about building, and about understanding. And, and really school didn't give me a reason. Why am I learning this? What do I want to know this for? How is this going to help me on my journey through life? And that's something that stuck with me. And so…

And yet this was a grammar school.

It was a grammar school. I made… I, I got the Eleven Plus.

Yes.

That wasn't, didn't seem to be a problem to me. But it was, I think, what I would call engagement. It was engagement with the curriculum. So I mean my technical drawing and my maths was, was OK; the physics was OK; my English was, very, not OK, along with my French, which was desperately not OK. Because all the things that were an adjunct to my making, I was interested in, and I took them seriously, and the things that weren't, I really let them roll. So my solution was, at sixteen, was to leave school, because that was the earliest point you could, you could get away.

You just left with some O Levels?

I left with some O Levels, yes. I mean my technical drawing was the top of the class. So, you can imagine, you know, how it goes down, and the, and the grade 9 fail in French was at the other end, and then they were on, on this spectrum between the two. Unfortunately my English and my, my English and English literature and language were just below the line. But that didn't matter, because I was going to go to a technical college, I was going to go to a technical college which, you could study… The course was an ONC in electrical and electronic engineering, at Bolton.

[08:21]
Right. Bolton Institute of Technology.

That's right.

Did this mean moving from home?

No. It meant an early start on a number 8 bus every morning.

Right.

And trying not to fall asleep on the number 8 bus coming back, otherwise I ended up in Manchester. Only happened twice. Once I had money; second time I didn't, and had to walk back from Manchester, which caused a little bit… I mean, we had no mobile phones in those days, so, parents were a little bit concerned. But still reasonably chilled when I told them what had happened.

Where is Peter? [laughs]

Yes.

[08:55]
You enter there in September '71, into the Department of Electrical Engineering. What was that course like?

Oh that course for me was brilliant. I think, the first, the first week we started in the, the physics. And they introduced me to the Pelton wheel. Not in terms of an equation, not in terms of a diagram in a book, but a physical, real Pelton wheel, there in front of you. And suddenly, those equations that I had struggled with in physics at school, actually come out of the page and suddenly present themselves in front of you, and you suddenly realise, right, that's what it's all about.

What does this thing do?

So basically, it's a, it's water, it's paddles, and then the paddles obviously rotate, and you can use that to generate electricity. And obviously the larger the head of water, the more pressure on the paddles, the faster it goes, the more electricity you generate.

Which is obvious when you see it. It's a set of bland equations. It's not quite so obvious what's going on.

You have to need the connection between the two.

Absolutely. I want, I want to relate things to the physical world. That's, that's what gives me the buzz. Because then I can see how that might be extended to create something else.
[10:20]
And of course, there, it was very early in, in microcomputers, and in fact, we, you know, the integrated circuit, microcontrollers didn't exist, but what we had was a PDP-8 made by Digital Equipment Corporation. Now that was the first minicomputer that you could get in a room that was, you know, roughly, what I call a standard size room. Like this one that people can't see, but you know, a few metres square.

Yes.

Whereas traditionally, computers had needed a huge room, needed air-conditioning, needed all sorts of things.

This was the first computer you saw?

This was the first computer I saw as something that you could get close enough to tough.

Yes. So this is, '71, '72, '73. This is quite advanced for the Bolton Institute of Technology I would have thought, PDP-8? Quite a bit of expenditure to buy?

Absolutely. I mean I think, although it's been hard to, to figure it out, it was something like £30,000 equivalent that they had spent on it. And because they had spent so much money on it, they put it in its own little room, with the teletypes. I mean, for those that perhaps don't go that far, interacting with computers in those days was, not even a VDU, which was obviously the intermediate. It's not the screens that we have now. It was something akin to a typewriter, an electric typewriter. And,

it just sits there and looks at you with this paper coming out of the top, and there's just this low hum coming from the motor that drives the mechanism. But I had looked, I had looked round the door, and I thought, I sort of, walked into the room, and the, the lab technician, who looked after it, said, 'Yeah, can I help you? What course are you on?' I said, 'ONC.' 'No no. No, sorry. This computer facility is for the degree students only.' So, my jaw dropped a little bit towards the floor, and I turned and walked out, and just as I was walking out he said, 'You can always come back at lunchtime.' And, his name was Graham Beech, and I remember him intently because, that interaction changed my life. What, what he did is, he got me back at lunchtime, he sat me down in front of this terminal, which just had one… So you've got to imagine, this is white paper and this one little arrow. So it's inviting you to type something. But what? I had no idea. I had never… Anyway, the inevitable program came out of the 10 print 'Hello World' 20 go to 10. And I think he had this as a little bit of a trick up his sleeve, and he says, 'Right, now in order to make that do something, you've got to tell the computer you would like to run that program. What do you think it should…?' 'Run?' 'Yeah, that's right.' Type in 'Run'. Return. Off it goes. And it starts spewing paper out, as it does printing 'Hello world' 'Hello world' 'Hello world'. And, after about, oh I'd say, 30, 40 seconds, there was a couple of feet of paper out of the top. Graham had reached for his sandwich box, and was eating his sandwiches, and didn't seem very concerned. I was starting to worry, like, how do I… So, 'stop'. No, that didn't work. 'cancel'. No, that, no, that doesn't work. And then I started eyeing up the, the wall switch. That's probably the only thing I could think of. You know, the worst possible thing you can do to a computer is turn it off at the wall. But that was the only thing I could think of. And, I think he waited for a bead of sweat to appear on my forehead. And at that point, he just leaned over and went, 'Control C.' And everything went quiet.

[laughs]

I mean, I had figured out that if I had been asked to work that out, even if I had known that you could hold two keys down at the same time, which, if I thought about it long enough, 'shift' for doing capitals, I'd have probably got there. There'd have been about 100 metres of paper. And, I think there was a massive lesson there, and that was, if you don't know, you really don't know, and the only way to find out is to ask.

I mean obviously today you can Google it, but, in those days it was to ask. So it made me realise there is no such thing in computing as a stupid question. If you don't know the answer, you don't know, you ask. The only thing that you have to be wary about of course is the stupid answer. But, I think that's massively important, because we need to encourage collaboration, which means asking questions, and admitting when you don't know what the answer is. And I think that then rolled through, you know, I would be putting my hand up saying, 'Sorry, sir, I don't understand that. Can you go through it again, or can you explain it a different way?' And so the level of engagement I had increased. And of course, having given me that access, that encouragement, I was back pretty much every lunchtime, trying to write bigger and better programs. Because the light bulb moment had happened where, if I could think about some algorithmic way of programming something in BASIC, then this computer could do it for me. So in fact, it absolutely appealed to the lazy side of my personality. You know, I can get this computer to do it. Brilliant. But then you could think, what else could I get it to do that I hadn't quite imagined yet?

[16:10]
You are two years there. And this man seems to be, to have set off this light bulb in your head, is much more than just a technician; he seems to have been a, a great teacher as well, a teacher by example. You quite like that don't you?

Yes, I think… I think the, the thing is, he was a teacher by encouragement. And so, you know, I've come to realise that, you know, your career is a journey, and that journey has hills in it, and it also has the benefit of the fact, going down the other side, you can coast along. But getting up that hill sometimes can be really really difficult. And students find that as well. They get to a point where it's really really difficult. That's the point where they just need a little bit of help to get over the brow of the hill onto the next thing. But, it's not rote teaching. It's teaching by, almost necessity. But the thing I think he did for me is, he turned around my educational career from being something, you know, have this, you need to learn this; have this, you need to learn that, to, I want to learn this. How does this help me with this? How does this connect to this? Tell me about that. Tell me about this. And that was the point at which it changed.

[17:31]
You almost changed to be someone who was seeking their self-education, is that right?

I… I think so. And I think a large part, a lot of people do. They, they have a pathway. It was just a question of having that goal. And my goal then, and probably still now is, to learn as much as I can about this wonderfully adaptable technology. But I see computers as a tool, always as a tool, to do something better elsewhere. I mean just a computer for computer's sake, not so interested. But, make it part of a system. I mean look at computers in cars. They've taken away a large part of the servicing requirement of cars because they can adjust everything on the fly to keep it running. And that's a huge benefit. And so I think, you know, my engagement has always been, a maker, computers, software, join it all together, and make something wonderful.

[18:35]
Your diploma was in electronic engineering, not in what would later be called software engineering.

No.

So, you were getting, through your lunchtime use of this PDP-8, the software side of it. And in the classroom and in the laboratories and the workshops you were getting the hardware side of it. Was that was happening?

Well that's right. But the electronics, you've got to remember that, that microprocessors really weren't on the scene then.

Sure.

So the electronics, the thought of doing the design of a computer in electronics there in transistors… I mean I've been very fortunate because Graham also, whenever the service engineer used to come to service it, he'd say, 'The service engineer's coming up today. I'll be back at lunchtime and have a chat.' I got a lot of information off the

service engineer about how you actually kept that PDP-8 running. I mean it had lots of gold-fingered edge connectors which were a little bit, they just get dirt on them, and they stop working. And he had so many techniques for finding out which card was giving the problem. And eventually, when the machine stopped, Graham would give me… Because he'd, 'Can you just come over and have a look, rather than call in service, see if you can see which one it is.' And when we can clean it, and we can put it back in, and we'll get it working again. And, again, it's all very physical side of, of that engineering. So yes, I, I started to look around, how could I learn more about this? And of course, I was very fortunate with having Manchester University just on the doorstep.

[20:13]
'71, the first Intel microprocessor. '72, Intel launches the 8008. The first pocket calculator launched by TI in '72. You are now in this tide which – well, tide, no, flood, which is just blowing out, both discrete products, discrete electronics, and also of course analogue electronics, and just putting digital and integrated electronics bang, right in the centre. And you are getting a big dose of that.

Well that's right. I mean, I obviously had to, you know, escape my course and get into Manchester University. Because they had the resources to provide access. I always remember the first 4004 board that they designed. It was, it was… All the support components were the size of a small planet, but right in the middle was this tiny Dill 16, 4004 microcomputer. And that was really it. And I think the real thing that took off for me is when we got hold of Z80s.

Right.

You know, which were a derivative of the, the 8080.

Yup.

And a lot of the development work that I did, I mean, the first computer that I built myself by hand was based on the Z80.

[21:37]

So in July '73 you got your diploma, your ordinary diploma.

Yes.

And you apply to Manchester University. You didn't have A Levels.

No, I didn't. I had…

You didn't have French.

No.

You didn't have Latin.

Yeah, no, Latin wasn't, yeah, wasn't…

Wasn't on their syllabus.

Wasn't a matriculation requirement for, for Manchester fortunately.

OK.

But English was.

But… And you got your English O Level literature and language while at Bolton.

That's right, I did, basically did a night school at Bolton to get them.

You thought, Peter, you had better get this if you want to carry on and get to Manchester, yeah?

That's right.

And it had to be Manchester, did it?

I… I… I think it did. I went to look at other universities. I mean, I had not, I had not really set my sights on Cambridge.

Right.

I thought that was probably, for a, a young northern lad was probably a bridge too far, but if I had thought about it now I would have said, you know, if had had somebody… I mean Graham obviously wasn't in the academic sphere. If I had had somebody who said, 'Oh, just go for it,' I may very well have done. And, if I get the opportunity I say to people, 'Look, if you want to do it, go for it. Figure out what you need to do, do it, go for it. You can do it.' But I mean, passing the, the O Levels was a, you know, it was a tense moment, because that would have blinkered my career. But I think this, it shows things doesn't it, that if you are passionate about something, then these little side issues you've got to deal with….

Absolutely. You've got to compensate for what you haven't got. And, if you're going to do that, you had better get on and, tick those boxes just to get them out the way.

Absolutely.

Your father didn't have a degree. Did he? He had a…

He had…

...practical background.

He had a totally practical background.

Right.

He had some qualifications in, in electrical engineering to allow him to certify bits of equipment. But he, he didn't have… You know, we weren't… My brother was the

first in, I think, four generations that had actually gone to university. He chose to do mechanical engineering. [pause] And, obviously, I was the second. And it, it had been tough, you know, because, we were very lucky then that we did get the grant, but the grant…

You got the grant. The local authority paid your tuition fees and also paid some maintenance to you.

Yes.

[24:18]
And indeed, you know, I went through the same process as well. Manchester University, Department of Computer Science. Probably couldn't have picked anything better, could you?

Not really.

Alan Turing had been there.

Yes. [laughs]

They were working hard on the Atlas machine.

Yes.

They had a close relationship with Ferranti, who were making the components and therefore also a close relationship with ICT as it then was, as ICL it became. And they had a man called Tom Kilburn there.

That's right. The critical thing is, they built computers there. And there was a really large hardware bias in the course content. It wasn't just programming; it was, computers, and computer-aided, computer design. How do we get the computer to help us build the next computer? How does, how does that work? And so, yeah, it was a magnet for me, because the, the laboratory experiments were all, you know,

generally, obviously we had the programming, but in the laboratory experiments we had bits of hardware to experiment with, and do projects on, and each year those experiments would get more sophisticated as more technology became available. And so, yeah, I, I probably, looking back, I don't think I could have picked a better place, even though it was on the doorstep, and I maybe thought I should have gone further afield. I mean, they, you know, they've been responsible for the Baby, the first stored program computer, and, Tom was inspirational. I mean I still read his paper on the one-level storage system. And, and you realise that some of the concepts in there have carried the test of time, they're still being used today.

[26:06]
Often they were working on, and you were working with them, being taught by them, and later on doing an MSc, and a PhD. And you, what were you having to do, I suggest, is to overcome the limitations of what the technology could do. Constantly having to do that. The limitations of the fact that, sometimes this would run for ten minutes and then collapse and it would take you a long time to know what had happened, whether it was a hardware or software fault, and mostly it was a hardware fault. The limitations of the fact that, you had so little storage then. So you had to come up with things like virtual storage, for example, which the Atlas machine had. So you're really having to overcome the problems of the technology itself.

Well that's right. I mean… I mean I came from, I came from the PDP-8, which had 8K of memory and ran five users of BASIC. So, you know, cramming things in to a, you know, a quart into a pint pot, was something that I, I had already admired that some people could do. But obviously, that's an expensive use of time. So if you've got more memory, great. So the, the concept of being able to have this massive address space, and you can basically throw things in half a mile apart, and then the computer takes care of the fact, well it really hasn't got that much memory, and it virtualises the pages that you're not using frequently, was just, you know, it was amazing to me. It was such a brilliant concept. Because it made the programming easier, you got this massive linear address space you didn't have to worry about. Yet, the actual mechanics of the physical implementation of that was subtly more complicated with the caches. And of course, cache memory is integral to what we do today, not just for giving you that infinite storage, but giving you that performance.

Yes.

And, they were always talking at , so, in the tea room and things, about, how can we change this cache architecture to make it better, quicker, faster, and do more with it? And, you know, it was just a fascinating time, and I absolutely thoroughly enjoyed being there.

[28:21]
What was Kilburn like?

Oh, he was… I never remember him raising his voice. I always remember him leaning back, and he had a very non-PC pipe then that he used to keep with him. He would always engage brain before opening mouth. He would, you would get this pause, he would think about it, and then he would tell you what he thought. And usually he was bang on the money. If we were having a problem with something, then he'd, he'd be in there with Dai Edwards, and they'd be sat round the table, and, they would brainstorm the solution with everybody. Basically he would collect all the information, think about it, and then, 'This is what I think we should do.'

Very disciplined man then?

I think so, yes. Yes, something I've probably not inherited off him. I can be a little bit of a live wire. But then when I… I normally now sit down, think about it. Probably leave it for a few hours, and then come back to it. Giving it chance to sort of digest in the brain what might be, what might be going on, what might be wrong with a particular thorny problem.

[29:35]
Talking about live wires, we're talking about the early 1970s, things were going, going at great steam in social and political sphere at that time. Were you engaged in that, or are you a man dedicated to electronic, electrical engineering?

No, I, I let, I let a lot of that go, go past me. The only thing I got involved with in the Students' Union was a thing that was called Contact Nightline. Because there were students committing suicide on campus, and you think, why, why, in such a wonderful place, why would you do that? And, I think we see that today, and it, it bothers me greatly that… I think that the education system is so pressurised, that failure does not seem to be an option, but, failure is, as we say in the foundation, is a first attempt in learning. If you don't… You know, it's like, I say to people, well, if it's dark, and you want to know how wide a road is, how are you going to find out? Well the answer is, you have to fall off both sides, then you know how wide it is. It's just, it's just one of those things that, that got me. So, I spent quite a long time doing that. And one of the benefits of that is, we worked with the Samaritans, they trained us up intensely. We were examined, we were challenged. And if you didn't pass muster, you weren't allowed to do it. But what it, what it taught me was that you could actually, not ask people questions exactly, and not lead them in any direction; they would find their own way, literally by just talking about the problem. And I use that in, in a lot of the problem-solving I do today. I say, 'OK, just tell me about the problem.' And the number of times an engineer has come in my office, said, 'Right, just tell me about the problem.' They get halfway through describing the problem, and then you can just see, sort of, ping! 'Right, it's OK. Got it.' And they walk out of the office. And I think that's been immensely valuable to me. But it was something that was left field from my, my computer science. But I just felt it was important.

[32:00]we'll write it down
By 1974, which is your second year at Cambridge, the UK had about… Sorry, not Cambridge, Manchester. I'll have to edit that won't I.

You will have to edit that. But you won't. I think we'll make you keep that in. [laughs]

No, we'll write it down. What's the number? 32:18.

[32:21]

By 1974, and your second year at Manchester, there are in the UK 4,500 computers. In Germany, 7,000, in France 4,000. And so on and so forth. So, we're still not talking about an immense amount of computing power. And we're still talking about a computing power looked after by the priests and priestesses, are we not?

Oh definitely. I mean in, in Manchester we had downstairs in our building the thing called the Regional Computing Centre, with a CDC 7600. And, you know, we still had that idea that, that computing was centralised. And obviously upstairs we had MU5, little MU5, and a few other adjuncts, and the PDP-8 that was in a, in one of the sort of, laboratory rooms. I used to go and visit it occasionally just to remember where I, where I came from. But it wasn't, it wasn't the era of the personal computer, but it was starting in the US.

It was. Because, in your penultimate year microelectronics had the advert from MITS for the build-your-own personal computer, and Microsoft was also founded to put BASIC onto that. And in the next year we've got the Zilog Z80 microprocessor. Did you say you used to work with that a lot?

That's right. So… So I… I was desperate to have my own computer, but obviously they were just, not available. So the only solution was to build your own. I managed to get a subscription to *Byte* magazine, which was one of the very early hobbyist magazines I call it from, from the US. But we also created a thing called the Manchester Computer Club, so I was a founder member of that, and we used to meet in the chaplaincy just right next to the computer building. And we used to talk about, you know, the idea of building our own computers. Some went down the 6800, the 6809 route; I went the Z80 route. But it was the camaraderie of actually building something unique for yourself. I…

Why did you choose the Z80? Can I just press you there?

[pause] You can press… I think the… I think the thing that got me is that, if you looked at the order code of the Z80, and I know this sounds, this probably sounds a little ridiculous, it had some missing order codes. And what they actually did fascinated me, and I thought, if I have my own machine I can actually investigate that

and find out. But I think the, I think the real… It's quite interesting. The real decision was based on the DOS. It was CPM. And I had access to a copy of CPM. And I could buy a copy of CPM. I probably shouldn't admit this on record, but I had an even better idea. I actually took the kernel of CPM, I dismantled it line by line, figured out what it did, and then rebuilt it. Because on my Z80 computer I had virtual memory. Because had learnt off Tom. I had paged memory. So, the memory, the 64K memory space was split into 16K segments, and I had something like, well eventually, I had 256K of memory on this machine. So I could create a virtual desk, which meant I didn't have to buy a second floppy disk. But it also meant that I could run programs bigger than anybody else, because, I could just keep a little bit of code at the top of the top page, to switch the pages out, and you could make a call to the routines, and the routines that live elsewhere. And, that was, again, it was a, it was another instance of, here's something. Take it apart, understand it, make it better. Which has really been sort of, the defining way that I've liked to go forward.

[36:45]
And, you took the next step, also within the Department of Computer Science at Manchester, by doing an MSc. And your subject was, 'Order code implementation in a high performance minicomputer'. Which was the MU5. What was all that about?

Yes, it was the little MU5. I mean it's not the big, the big one that had been done in collaboration with ICL. It was a, sort of a mini version of that, a satellite if you like. Well the interesting thing is that, you know, I was very interested in computer architecture, and how instructions in memory get executed. And there are so many different methodologies for doing that. And I thought, this is really, you know, I'm really quite interested in this. So, I can do an MSc actually researching this. And also figuring out how to do it well in, in a, a new design that we were doing. And, eventually I settled on a structure that basically, you would get an instruction in that would be decoded, essentially in another computer inside the… And it would spit out instructions to all the various units to get them to do the things in synchronism to actually make the, the computer do its thing. And then I realised with that level of power there, I remember for one open day I actually wrote programs in the microcode of the computer, because, you know, because you could do that. I thought, I'll just write a few programs, and I wrote a few games that people could play on it, written in

the microcode, not in the, the, what was the, the high level language that you would traditionally have in the main memory. And I was just fascinated by this, this sort of, layered architecture that we've got. And if you actually look now, I mean it's even more layered isn't it in the way that the computer systems are built. And, and that, that attracted me. And I did enjoy that, and we were successful, it did work. So I was very happy with that. And then went on to tackle the next major thing, which I think you alluded to, is, when something doesn't work, how do you find out what's broken?

[39:01]
Who was your supervisor?

It was… [pause] Linda Brackenbury. And she was a great inspiration. She was always very positive when we had, we had problems. We had had one instance when, a major setback, where we had built a load of the boards for the machine, and, we had sat there on the bench trying to commission them. And eventually the light dawned. Because we took one off the board, we metered all the pins, and got no impedance between any of the pins. This can't be right, we said. Well there's only one thing for it. So we ground the top off, only to find for some reason which has never been explained, there was no silicon in those chips. It was just a blank fabricated package with no die.

[laughs]

And there was, only about 40, 45 of these in the machine, but we got them all out. Put working ones in. Bingo.

Who had made them?

I, I couldn't possibly comment. That would be, that would be unfair. That would be unfair.

[laughs] But you did have a close association with Ferranti.

Yes, we did. And it wasn't them.

OK.

No, it was a, it was a major, a major US manufacturer, that had a little bit of a slip-up on their production line.

[laughs] Oh right.

Because it was still early days for all this technology.

Yes.

And also, the packages were, were still fairly huge.

'76 to '78 you did this work, yes?

That's right. But then again, that's probably what drove me on for trying to figure out about how you could test a machine. And so I had started work on scan chains, the thing that became JTAG in modern microprocessors. I did a lot of work on that, we got the computer running, I wrote all the code, the software. Proved all the hardware worked. And I was doing this for a PhD, but of course the big thing with a PhD is, writing it all up. And, fate unfortunately intervened at that point. I had made a video frame store for a, again, for an open day. I loved the open days, because it meant interacting with the kids and the public, and trying to engage them. And I think that was a, sort of a, a signal of things that were going to come in the future. But, they… I made this. And, a guy came into my office about, a week later, and said, 'You know that machine that you made that displayed images?' So it used a camera, put it into a frame store. Turned it into false colour, and then displayed it on a monitor. Now those days that was, that was a fairly cool thing.

Mm.

I mean nowadays we just laugh a lot about… But, you know, you have to put it in context. Always have to put these things in context. And, I actually had to go down

to the University of Manchester computer graphics department and borrow a colour monitor, because we didn't have one. And so, he had seen this and said, 'I'd like you to…' and again it always sounds crazy, 'I'd like you to use the technology you've done here to design me a T-shirt machine.' Now, a T-shirt machine then, what it meant was that, somebody would stand in front of the camera, it would take an image of them, then it would print on a piece of Mylar, heat transferable ink, in their image, and then they would iron that on the front of a T-shirt. The sort of thing you would see at the seaside. And they still exist. Obviously much more sophisticated now, but they still exist. And I… And that sort of caught me unawares really, because that was the first concept of commercialisation for me. So, I used my normal idea, well, why not? And I can't think of any reason why not. So yes, sure, I'll do it.

[43:02]
So after nine, well eight years of, of academic work, from Bolton through to Manchester University, you decided to leave and not complete your PhD.

Yes, I didn't, I didn't complete…

Didn't submit.

I didn't submit.

Regret that?

Um… [pause] It's a hard one. I think, the answer is, yes and no in equal measure. Yes because, I should have put my grit to the mill and done it, and I really respect people who do submit their PhDs, because I realise how hard it is, and how, you know… And, so, yeah, on that side I regret. On the other side, it would have probably closed another door. And this other door looked more inviting to the maker side of me.

It was the maker side of you.

That…

And you'd probably make some money, would you not?

Yes, I would make some more money, which is always something that, that you have to think about, I think, ultimately. That, that putting food on the table is quite an important thing. But, as well as putting food on the table, it gives me some cash to start spending on other things that I'd like to build.

Did you have to put food on the table purely for you by now, in '79?

Yes, by then it was just, it was still, it was still just for me, but was, you know, very soon to become with a wife.

Yes.

So, I had, I got married, so we had to have a house. And then we moved on there. So yes, then there was some sustainability. She was doing a, doing a PhD in concrete.

Concrete.

Yes. And, the mechanics of rebars and how rebars corrode concrete, and how you might actually measure. And, mitigate.

Where was she?

Again, that was at UMIST actually, just so… Because we, we met at the university.

Oh right.

But then…

[45:14]
While you were… If I could just backtrack a moment.

Sure.

While you were an MSc student, and while you were, also started your PhD, you were teaching as well, were you? Because they roped you into teaching. Did you enjoy that?

[hesitates] I actually loved the teaching. I even loved the teaching where, one slightly, [sighs] cantankerous person had taken a page out of my notes, and left a blank piece of paper with a little smiley face on it. Most, they were most disgruntled, as I just wrote straight past it, because I had learnt all the material in advance and I just used the notes as sort of, as a, as a sort of a, check that I was going in the right direction. Because I have a terrible habit of wandering off into some backwater of interest. But I did enjoy it. I enjoyed it most when I got to the MScs where they would actually ask questions, they would challenge. It would become a collaborative piece of teaching rather than a rote piece of teaching. I was also very passionate about keeping the notes up to date, trying to make sure that we were teaching the latest and greatest about, about what was going on. Which is a lot of work. And that, and that to some extent, you know, meant that my MSc was a little bit late in submission, and probably was partially the thing that was the death knell for my, my PhD. But I, I did enjoy it. And so, moving to the dark side of industry was going to be a big change.

[46:55]
And so someone came up and said, 'We've seen you do that on the open day. Can you do something because we want to print T-shirts?'

Yes.

And you said, 'I think I can.'

I… And I did. Unfortunately, at the point at which we had a working prototype, their backers disappeared into the night, and the project folded. But, it just seems to have been a feature of, of my career that, out of, out of adversity, something successful came out. The, the company that I was working with also were involved in video security, and recording cameras to tape. Now, when you are recording cameras to

tape, then, if they're standard video cameras, they're asynchronous, so you don't know where the frames of data are going to be. They're not aligned. So what you have to do is, run an extra cable out to every single camera to say, 'You need to be in sync with this signal. You need to be in sync with this signal.' So we're all singing from the same hymn sheet when we record them to tape. Because the, the way that you saved money in that environment was, you would record, eight or sixteen security cameras on the same piece of tape, and you were progressively just writing a frame from this camera, this camera, this camera, this… Because, you didn't need them all to see what was going on. And we realised that this frame store could be adapted to save having to run that extra wire, and could be used to, if you like, cue the images to go to tape. And so with a little bit of fiddling, we could actually, we could actually make that work. And, so we then quickly created a product, and, it wasn't quite… It worked. It wasn't quite what we wanted. And then, a couple of years later, we created another one, which we called Uniplex, and that was what we wanted. And again, that had a component of my tinkering. The engine that drove all the graphics and controlled it all, was a microcontroller, display controller, that was used on 6800 systems, and 6845 if I recall correctly. And I had looked at the datasheet, and I said, hm, it doesn't say it needs a clock between these two frequencies to work, because it's a dynamic, because it, dynamic designs were cheaper then. It doesn't say I can't mess with the frequency. So I figured out how to mess with the frequency, and I stalled this chip until the camera data was ready, then I'd start it off. It would generate all the signals to put all the thing in the frame store. And, it basically cut, 30 per cent out of the bill of the materials. And then I found I could use one on the output to do the same thing. And then, there was this new technology on the market called, the GALs, which were, or PALs, which were program array logic, which were quite simple, but basically, you had a chip, but what was in that chip was your choice. It's a precursor to the FPGAs really that we see now. And, we started to use those. And that reduced the bill of materials. So the price point on this was really competitive, and it sold really well.

[50:21]

And you were based in Manchester; this company was based where?

They… They were based in Manchester. My company was based in my back bedroom at home.

So now you had formed your own company.

I had.

Called?

It was called Mosaic, which was a great name, up to and including the point at which the Mosaic browser appeared on the Internet.

Oh yes.

At that point we went from being in the top four to being in the bottom four [laughs], overnight, in searched. So, that was unfortunate. But, we had already established, had established a market.

Were you working on your own? You were working in collaboration with another company, but in your own company…

I did.

…were you working on your own?

I was initially working on my own. Then I had recruited another member of staff. He shared the same passion for getting things done. Eventually he became a co-director, then we recruited more people. I then took… The back bedroom was not, didn't give the right impression. We moved to Carrington Business Park, took a little office. Then took the office next door. Then took the office next door to the office next door. Then took the wing. Then… Almost took the block. We were almost at the verge of taking a block of the offices. And then, I got a phone call saying, 'I don't know how to say this, but I need to buy your company.' And I thought, well, there the strange

phone call, but, that's fine. And after a, a bit of wrangling, we sold that company, and started again.

That was '86, you sold it.

Yes.

And started again.

No, it was ninety… It was '97 when…

'97.

[52:12]
'97. So it was '96 when the call came and '97 when we actually sold, and created this new company, Norcott Technologies, that I'm with today.

You've got a big challenge here, which is, now you become a manager.

Yes.

Tell us about that.

Well, I think, there is quite a bit of a difference between a doer and a manager. And, it had always been a little bit of a struggle for me, because I enjoyed the making and the doing. So in Norcott I actually went on a course at Lancaster University. You see, although I had…

But even before at Mosaic, you had had to manage.

Yeah. Yeah, I, I managed, and that was, that was OK for, for reasons I, I can't explain, that just… I think because, pretty much everybody were engineers. I could set them on a path and a target and a direction, and basically leave them to it. There was no a lot of sweeping up. What changed was that, we had built all our own

prototypes. We had a lab. We just sat there and we just, got somebody to build them up. And it was easy, because you could hand solder pretty much any electronic component. But the advent of surface mount was making our life, or starting to make our life miserable. Because, you had to start sending stuff out, and you'd have to wait two, three, four weeks for it to come back. And, that wasn't comfortable, because we, you know, we're eager to get on. So, in, when we moved to Norcott, we bought our own surface mount line, which would then sit there for, weeks on end doing nothing, then there'd be a huge hubris of activity whilst we built prototypes for a product, go upstairs, get them, get them commissioned. And that obviously isn't a sustainable business, but what very quickly happened is, people said, 'Oh well you've made the prototypes, could you make the production for us?' And so, the business morphed into a hybrid of design and manufacturing.

[interruption]

All right. Sorry about that. Just need to need to check the time and just to…

Fifty-four. Seventeen. Chuck that out. Yeah, they were going into that room there. It's a bedroom, would you believe.

No comment. Yeah.

[54:50]
So your company now at, called Norcott, n-o-r-c-o-t-t…

Yes.

Norcott Technologies, based in and around Manchester.

Well we're, we're based in Widnes now. We were, traditionally it was, we were in, nearer Warrington.

Right. It's both a consultancy in terms of design, and you are designing, all type of applications, of digital control and computers, and also now a production line for other people.

That's right. We, we generally… So we, we manufacture products for our customers, most of which have been acquired through our design side, not all. And we have, we have three, we have three SMT lines, it's quite a bit mouth to fill. But, the reason for that, in my eyes, is that, if we make it, we can then observe it being made, we can learn how to make the design process better, we can close that loop. We can say, ah, OK, that isn't quite right, that's not soldering as well perhaps as it should. We can improve the design there. And I can actually get my engineers to go and observe the difficulties that they may have caused in the downline process. So again, they understand, and then they won't make that, that mistake again. And it's all, again, about optimisation and doing things better. And we do a lot of industrial and scientific. I love getting involved with the scientific things. We build things for CERN, and that's always fascinating, because they're always a bit… I don't… I don't know whether electronics is allowed to be funky, but it, it's just… Because it's a special, for a very particular application, like the Large Hadron Collider, it's got features that you wouldn't see in sort of, normal commercial products. And so I, I love all that sort of stuff.

How many people are employed?

We've got, 64 now.

Still quite a tight company.

Yes. You know, we're very compact. I think, going back to your question. So I went to a, I went to Lancaster University on a course about small companies and SMEs, and particularly owner-manager businesses, and, it was really there they convinced me that, if I'm passionate about the engineering, why am I Managing Director? Why don't I give that role to somebody who's perfectly capable, and probably actually more capable of being a managing director than, than I was? I'm not, I'm not ceding

control of the technical things, the things I'm passionate about, but, it'll probably be a happier life. And they were dead right.

So you became Technical Director.

Yes. Or Director of Engineering as I, I chose.

Director of Engineering. I'm so sorry sir.

I mean… I mean it's, it's quite right, we, we choose, we choose these titles, we choose titles. You know, I would have been perfectly happy with, the guy who designs stuff. I, I don't, you know, I won't sort of, acknowledge title… You know, I can have… I mean just the other, the other week, we had an apprentice coming in from college, he has worked in software. I set him a task, along with my IT team, I said, 'Right, look, we've got this stupid book for doing GDPR, sign-in, on the front door, so people can't see who else is here. I hate it. Can you do something, can you write me a piece of software and put it on a touch screen display?' And within, five weeks, that was there, and done. And I was thinking, well why did that happen? Well, for exactly the same reason that I learnt to program. Because, he came in, he was given encouragement by me to go and do this thing. The IT team were encouraging and helping him learn all the things he needed to do. And, he achieved a goal. And, that's something that, that you know, we, we may talk about later, is the thing about the physical electronics with the kids, is that, if you can get them to a point, even if it's a, LED where they write the code, they connect up the LED, they then decide to connect that LED up a little bit differently, and it works. There'll be, probably, might be some tears and some pain before that. It's a very important, yes, I did this, I wrote the code, I made it do this, and it's mine.

[59:39]
Where did you get the money from for Norcott? Was it from the sale of Mosaic?

Yes. Yes.

OK.

And that was…

Did you have to go to a bank or anything?

No. I've never, never borrowed money off, off banks.

OK. Never borrowed money off banks. And so you were able to chart your own course through until a, a moment some time in 2007 or 2008, where you made a momentous decision which has been of help to hundreds of thousands of, of young people, which is to develop a computer called Raspberry Pi. Tell us about it.

Yeah. I don't think… I don't think I made the decision; the decision made itself. So, we had built… At Norcott we built a board for, for Imperial College. And, Professor Wayne Louks phoned me up and said, 'Do you want to come to the open day and see it working?' And I, oh God, it's London. Big smoke. Oh go on then, I will. And, I was sat next to, to a chap, I didn't know who he was. I was never that good in the social situations, going round asking everybody's name and asking what they did. I think, well, if I need to know that, I'll ask them, but if I don't need to know that, I, I don't feel that I should invade that, that space. Anyway, we got talking, walking over for lunch, actually in Hyde Park, I had to walk across Hyde Park for lunch. And, and it turned out to be Alan Mycroft, who was Professor of Computer Science at Cambridge. This place, I didn't think I would ever talk to somebody or even go there. And he was talking about this, this idea for a little computer to reengage the students in, in practical programming. And, he was talking about it, and I said, 'Oh, can I come down and, and meet you guys? I'd like to talk a bit more about it.' I went down to Cambridge in the October for a meeting, and literally, within, within half an hour I said, 'We, we've just got to do this. I don't care, we just have to do this. It's such a good idea. It absolutely solves a problem.' It made my eyes, I mean, that…

What was the problem?

OK. For me, and I think for them… So in electronics the problem was, is that, kids weren't tinkering with electronics. For them, kids weren't writing software. They

had become consumers rather than creators. And so that, as that followed them through their career when they applied to Cambridge, then, there was a pronounced lack of computing skill. I think, Eben Upton quoted once, he reckoned it took 20,000 hours to become a competent programmer. I worked that out, that's quite a long time, you know. It probably takes 20,000 hours for my son to become a competent game player on some of the new games, but I'd rather, you know, I'd rather he didn't, but, we just meter it out, it'll take him a while. But, we've seen in electronics that, if I showed a board to a prospective candidate, they couldn't identify the components, because they had only dealt with them on paper, and in simulation. They had not physically got their hands dirty with them. And I make this rather rude comment that some of them probably didn't know which end of a soldering iron was hot. I mean they'd rapidly find out, but… And so that was a problem. And I saw this as a potential way of getting back to a situation where they could do this. Because what I felt had happened, and I, I think we all concurred with this, is, electronics and software was becoming packaged. The advent of the, the PC, which had really awkward and clunky interfaces to the outside world, had started to close down accessibility. Whereas with the BBC Micro, you had all that accessibility, you have this interface, you could connect things to it. And it's also a vibrant community, to help you do these things, and overcome the problems. And then you got, I don't know, I don't… I'll preface this, that I don't have anything to say against Apple, but as an example of a closed system, it's almost perfect. So, the key thing there is that, if you were curious, as a kid, you couldn't really get involved with it. I mean you're going to take the back off your MacBook Air and try and put LEDs on it, a) because you'd have no idea where to start, because it's just too complicated, and b) even if you did, you wouldn't be able to program them. And also, it's incredibly expensive. So it's high risk. And Alan Mycroft, you know, talked about this idea of activation energy, that you have to bring the level down. I mean I, I, I equivalent it to trying to do the high jump at school. If you put the bar at eight feet and say, 'Jump over that,' and you've never done it before, you're going to go, 'Really? Nah.' Not even going to try. Impossible. If you put it at, two inches off the ground, why am I going to bother with that? It's obviously trivial. Not interested. You have to find a middle ground where there's some amount of difficulty, and cost, but it is worth having a go. Oh yeah, mm, yeah, I think, I think I can do that, and have a go.

[1:05:06]
Did you have a price point?

Yes, we did. The price point that we, we ended, well, we ended up with, was 25 dollars and 35 dollars. There was a quote that it should be the price of a textbook, but clearly, the people who were pricing it hadn't looked at a textbook recently, otherwise it would have been 70 or 80 dollars. But no, it, it seemed, it, it's the right sort of, ballpark, because it's under the threshold where people are prepared to take a risk with it. So if you blow it up, you know, the worst that could happen, then it was only going to cost you the equivalent of a meal for four out to replace it. And for something that was going to advance your learning and your skills, you could see that as being worth it.

[1:05:54]
And you not only took that decision, with others, but you also decided that, this isn't going to necessarily make Pete Lomas the next Bill Gates, or whatever. You turned it into a charitable foundation. Why did you do that?

[pause] I think… I think the decision there was, one of mission. I, I would have found it incredibly difficult to say, 'We're here to help with the education of kids. Buy this computer, and I'm going to make a load of money out of it. Thank you very much.' It just, didn't sit well. And on the counterpoint to that, you know, we had got our own businesses, we had some, as I said before, I've got something that puts food on the table. So that wasn't the concern. What… There was another underlying reason, which is probably more, much more important, is, it takes me back to Graham. And, what he did for me, the opportunity to play that forward, with other students, to give them access to a computer that was low priced, it was, that was the, the thing that was most important. And so making it a charity, seemed to me to be the right, and, and to everybody, to be the right decision. So that the mission was very clear. It wasn't to make money; it was to invest in the future of society.

[1:07:33]
Is it light on software?

[hesitates] I don't think so. And the reason for that… I mean, the original…

Because I looked at the spec.

Yes.

And I thought, a bit light on software. I can see a, a hardware man coming through here in this design, and I wondered whether you thought now, in retrospect, it was a bit light on systems software.

OK. I mean I think this comes in, this comes in two parts. The first part is, yeah, I mean the two… You know, the, the two people that burnt the midnight oil on the design of Raspberry Pi, it was myself and Eben Upton. He's a passionate designer of chips, and, and that technology, and he, he was working in Broadcom. He also had a history in games. I don't think either of us were massive software, certainly wouldn't call myself a software guru. And so yes, there's an element of that. But I think, the decision to get to a point with the BCM2835, which gave us Linux derivative, a large part of the software is basically, whatever you want. It's available, you can download it and you can use it. So that, the ability to hug close all that brilliant collaborative and community work, well, why do we need to try and replicate that? So yes, it looks light on software, but in fact, if you look at it, it's extremely rich. The other thing that it has, and this was a terrific part of the, of the project, and I think makes me… Because we were a charity, we started to get an army of supporters behind the mission, who helped in, in so many different ways, putting in time, effort, some cases money, or writing resources, or just supporting us. And if you look at the breadth and depth of support that you can get for the Raspberry Pi, that's what's made it also a success.

[1:09:52]
How many are out there now?

I think, at last, last official count, we were rolling around 28 million, but if it clocks up at the normal rate, we, we must be in, we must be in touching of 30 million in the next, you know, few months. Which is extremely gratifying.

And it's international.

Absolutely. They're all, all over the world. Again, you know, I think the thing that has helped us is when we initially, we initially took on the decision, we got to a point… I mean there was a, a real sort of, awakening moment, when we asked how many in our community would like to buy one, we were expecting, we probably wouldn't make the, the 10,000, we wouldn't get 10,000 orders for the 10,000 we were making. 200,000 people put their hand up. That then gave us another commercial dilemma of how we were going to get, going to service the need. It took us a while, but, with the huge support of what was then Farnell and RS, who actually waded in and agreed to help us with the mission, it worked. And again, I think that, that's another thing you're thinking, if we had gone and said we were a commercial business, we want to license these designs to you, we want you to make them, we want you to put your logistical skills behind it, to actually make it achievable, I think if we, we had been a, you know, a limited company, they'd have probably gone, 'Well, maybe not.' But it, it really, it really did appeal. They saw the mission, they were on board with the mission, as our current manufacturer, Sony, are hugely passionate about the mission of democratising this technology by making it cheap enough and accessible enough that anybody can have a go, and providing the resources and material around it, so that they can have a go with knowledge and support. Sometimes that didn't exist when Graham was there, but I mean he, he gave me that access. And then, having a community, and having our outreach programmes, that allow us to then do essentially millions of times what Graham did for me, is, is extremely gratifying.

[1:12:15]
What's your biggest mistake you've made?

Well that's a horrible question.

It is.

That is a horrible question. [pause] On Pi, the biggest mistake I think I made is forgetting to connect one of the power supplies up, which meant that, the first prototypes didn't work at all. That was a, an unbelievably easy fix, which for a, for a complicated piece of circuitry, you know, was an instant worry. It's one of these where the, you know, where the electronics rearranges your bodily organs and your heart drops somewhere below the knee momentarily. Until the engineer in you kicks in and say, 'Right how do we fix this? What do we do?' Mm. But… Yeah, no, I… I've never, I've never really been one at looking at mistakes. I mean lots of people say, 'Oh your, your mistake is not taking a cut of every Raspberry Pi.' And I go, 'No. No no, that's not a mistake. That's absolutely, the right thing to do.' Raspberry Pi has rewarded me in spades. We go… Our CEO, Philip, said to me, 'Oh look, I've been over, I've been over n Ireland and I've been to this thing called Coolest Projects. You really need to come this year.'

It's called what?

Coolest Projects.

Right. Coolest Projects?

Yes. And it, it's one of our, now one of our team, you know, part of our, the Pi family, called CoderDojo , and they ran this, this sort of like, competition weekend where, the kids come together and show off their projects in all the different disciplines, hardware, software, apps, games, and we have a judging panel and they round, and eventually there is, there is the key winners. [pause] And, I was absolutely blown away. The amount of innovation that these kids, you know, young, as young as ten and eleven, were actually demonstrating, was fantastic. I mean, OK, some of them, the projects were a little bit immature. I don't… And I don't mean that in that they weren't well thought out; it's just that the technology was not available to them to do the complete thing. But talking to them, and the passion that they had for their projects, the knowledge they had about their projects, their understanding, I thought, yeah, this is, this is what… And of course a lot of them used Raspberry Pi's. So, yeah, this is what it's all about.

[1:15:02]
Can I put a philosophical issue to you…

Sure.

…towards the close of our, a very interesting and inspiring contribution that you are making. Isn't there just too much computing? Isn't there just too much technology that we cannot, although maybe your users of Raspberry Pi can, but the ordinary people just cannot control, and that it is beginning to take control over people's lives? And indeed in the medium term, there is the threat of AI, where some people who I have interviewed, Sir Clive Sinclair says, AI? Forget humans. It's AI. It's more intelligent. It's going to take over. You've contributed a lot to that, haven't you?

[hesitates] Contributed a lot to the, yes, to the problem maybe, yes. I mean you can put a, a foundation AI system on, on a Raspberry Pi. But that means you can tinker with it and get a better understanding of how it works. I mean I'm not, not suggesting that anybody really knows how some of the decision-making processes that AI actually makes to actually achieve its decisions have got any level of clarity. I've been amazed about some of the things that AI has been able to achieve, for the positive benefit of society. Like being able to diagnose, you know, cancers where they would be missed by a human. Humans are good at being creative, and humans are good about innovating, if we let that, if we let that come to the fore. I think that the problem… The problem that we've really got is that, you can use technology for good or, financial evil if you like. And, if you have no idea how it works, you have no idea how to get control of it, you have no idea of understanding the implications of saying, 'OK, I will have my house controlled by this thing that works off the cloud, that works off a server, somewhere in the back of beyond.' If you're aware that if that server goes down, then your house goes down with it, and all, you lose the control, you're going to say, 'No, sorry. I don't want that. But I want one that has a local intelligence, and I can talk to the cloud if I want to, but if I want to cut off that connection, it still has to work.' And it gives the consumer then, more information. They're more informed about what they're getting into. But I think, you know, you're right, the pace of change is so great, that, this technology, Wavefront, is accelerating away from the understanding of people. But if we don't try to give that

democratisation of that technology back to the people, in what we are trying to do with the kids, then it's going to get even further away. And then we, yes, we probably are ending up in a AI Armageddon of some sort. And, and I'm not sure, I'm not sure I can tell you exactly what that would look like. It may be just something as simple as, an infrastructure just collapses, because we've let AI run it and then all of a sudden it goes wrong. And nobody has any idea of how to fix it.

[1:18:35]
But it's probable, and, I think you are saying, if I can paraphrase it a little bit, it's probable that the people who could fix it have actually learnt such things, because they had a Raspberry Pi when they were young.

Well, [laughs] I would hope that that was the answer. I think it's… I, I think it's more that they had a Raspberry Pi when they were young, they became interested in it, they became passionate about learning and understanding. And then that drove them on through their career to go on and learn about things that are more, that are faster, better, more interesting than a Raspberry Pi, dare I say it. Because, it's, it's something to get people on that journey. And when that journey turns into the 20-lane motorway, and it goes in all different directions, it's for them to choose which way they want to go. The Raspberry Pi just shows them a bit of that road, a bit of the opportunity: here's a bit of AI, here's a bit of robot, here's a bit of, just doing some straight computation, here's a bit of doing IOT. And I think the thing that has really inspired me is that when you looked, last year's Coolest Projects, so many of the projects that the kids came up with were for the benefit of society. They were trying to make a positive, either environmental or social benefit, in their, with their project. And, and that was tremendous. And I think that bodes well for the future.

[1:20:14]
So you left in about '79, 1980, from the academic world. Didn't finish your PhD, but you did get a PhD, did you not? You're a Doctor of Science from the Manchester Metropolitan University.

I did. I, they very, they, you know, I was very honoured that they, they recognised my work with Raspberry Pi, and gave me an Honorary Doctor of Science.

And also, Peter Lomas, you've been showered with awards, from various institutions, for your work particularly in Raspberry Pi. Thank you very much for your contribution to the Archives today. It was meant to be an inspiration, and it will be an inspiration. Thank you.

Thank you.

[End of Interview]