Oral History of Massimo Banzi

Interviewed by:
Shayne Hodge

Recorded: May 12, 2015
Mountain View, California

CHM Reference number: X7483.2015

Shayne Hodge: Hi, today is Tuesday, May 12 [2015] at the Computer History Museum. We have with us Massimo Banzi, who can correct my pronunciation of his name.

Massimo Banzi: That's pretty good.

Hodge: Massimo is best known for his co-creation of the Arduino hardware and software ecosystem, and so welcome to the museum.

Banzi: Thank you.

Hodge: Perhaps we could just start with your background. Before you created Arduino, what was your educational and professional background?

Banzi: I'm going to summarize, but it started off with-- when I was a kid, I was taking apart everything that I could put my hands on, so people understood that I was going to do some kind of engineering job. And I started to get really into electronics when I was eight because I was using this kit that was made by this German company Braun-- or "Brawn," some people pronounce it --where you could snap together these cubes magnetically and build circuits.

So for me as an eight-year-old kid, it was amazing. 15 minutes, I built a radio. It was also important for me because the person who partially designed that kit was this German designer called Dieter Rams, which was very influential in the design that Apple does now. So in a way for me, it was an introduction to electronics and an introduction also to an interest in design, which became more important later on.

Then I studied electrical engineering. I went on to university to study electrical engineering. I never finished my degree because I was bored. I don't know. I was already working after school building projects, software, hardware of different kinds. So in a way, I sort of lost interest in the university.

And then the web started to happen around '93, something like that. So I started to build the first websites, and I went to work. And I worked for many years as a web developer, software architect, doing web projects. I lived in London for awhile. And then, I had one year where I worked the CTO for a venture capital fund, and after that one year, I decided that I needed to do something different with my life, and I always liked to teach.

So I decided to start teaching. And that's how I sort of at some point ended up being-- I became an associate professor in a design school in the northwest of Italy, which is where Arduino happened.

Hodge: So what design school was that?

Banzi: The school was called Interaction Design Institute Ivrea. So to give a little bit of background, Ivrea is this small town in the northwest of Italy. They used to be the city where Olivetti, the company Olivetti started and where Olivetti went through this whole kind of history where it had also some important peaks. It was quite important in a number of ways.

And then, in the '90s, the classic computer business of Olivetti started to fall apart. So in a way, they went through massive restructuring. They laid off a lot of people. So one of the various things that the new owners of Olivetti wanted to do was to create some kind of a school, a master program, that would be part of their way to give back to Ivrea after letting go of something like 100,000 people overall inside of Olivetti, not only in Ivrea.

And I think the story as I know it is that they originally wanted to do a business school. But then, there is this woman whose name is Barbara Ghella again who was involved in the project, and she was one of the

first Interaction designers in Italy, so people who designed interfaces for software for devices, and we're talking late '90s, beginning of the year 2000. It was very early on, and she knew a lot of people around here in the Bay Area. She knew people in London.

So she said, this new discipline of combining design with technologies, using design to design the interface, the way people interact with the devices, not just the shape of things, is new. We should do that, not a business school. There's a bunch of them. Let's do that. So they actually did it. And they hired the director of the Interaction Design Program at the Royal College of Art. They had a bunch of people coming and mentoring the school, so one of them was the co-founder of IDO, who died recently.

Another one is Bill Verplank, who was a professor at Stanford. So a lot of people that were relevant joined the school, and that is a kind of design school which is not the classic Italian design school, which is famous for designing beautiful tables, beautiful lamps. It was more modern, more in tune with what was going on in the tech world.

Hodge: So what year was that founded?

Banzi: I think it was founded technically in the year 2000, but it became really operational in 2001. And then at the beginning of 2002, they were looking for a faculty member that would be able to teach electronics to students. Because again, one of the key things about this school was this idea that if you design a new type of device, a new type of technology, you need to be able to prototype it.

And since the world was already going more and more into devices, from computer to more hand-held devices in a way, you need to give students the ability to prototype with electronics, which normally requires a degree. It requires years, but some pioneers in the field already had figured out that you could use cheap microcontrollers. And you could teach students in a matter of weeks enough knowledge that you could really build the prototype of a small device or something.

So there was already some groundwork done by Durrell Bishop at the Royal College of Art or Dennis Sullivan, Tom Igoe at ITP in New York, and there's a bunch of other people. All these people already started in the '90s figuring out that you could teach electronics to a designer by taking advantage of these cheap microcontrollers, simple programming languages, shifting from analog electronics, which is complicated to explain to a complete beginner, to microcontrollers, where a lot of the responsibilities shift to software, which is a little bit more manageable when you're teaching.

Hodge: Was it hard for you to get appointed as a professor at a school when you had said that you hadn't finished your initial degree?

Banzi: Design schools, in a way, they're interesting because they're based a lot on practitioner teaching. So a lot of design schools are less academic, and they're more about doing things, being somebody who works in a certain field. So a lot of designers who are designers by profession teach in schools. So it's easier to be appointed as a professor if you can demonstrate essentially that you have some kind of ability in the field, that you are part of a certain field.

And I think for me, there was part luck. In a way, I was available. I really wanted to do it. I wrote a beautifully long and elaborate letter, a motivational letter, where I wrote to the director saying, I would really like to join. This is like the new Bauhaus. We're doing some amazing stuff. I would really love to be a part of it and my feeling has always been that she wasn't really convinced. But I was available, so I started to work with the students.

I think I had an interesting interview the first day I visited with Bill Verplank, who was in town from-- he was in Ivrea visiting. And in a way, I think he was a little bit more convinced. The director was less convinced because I was lacking a specific degree in interaction design or whatever. But I was available.

I started to work with the students, and after a couple weeks working with the students, they liked working with me, and we built some interesting projects.

And again, I think it was very useful for me the fact that most of my learning in electronics happened first as a kid exploring. And then when I was in school, I sort of learned the theory later. So I was able to teach the students the same way I learned as a kid, so learning by doing, project-based learning, really not spending too much time upfront talking about electrons, atoms, electricity, and voltage or whatever, but by actually getting people to build stuff and then developing a method which is based on them asking, please explain to me the theory now because I need it to go to the next step.

Hodge: When you were working as a software programmer in the web world and then later as a CTO with the venture capital company, were you still doing hardware designs on the side to keep current in the area, or were you doing any hardware work at that time?

Banzi: Not that much, actually. I have to say that for a number of years, I posed a little bit. In Italy, we have this thing they call radio amateur fairs, but effectually, they're these hardware swap meets. Radio amateurs get together, and they swap parts, and they sell parts, and there's people coming and selling surplus parts.

Slowly, it went from radio to computers, so people started to actually have computer parts. I did a lot of-- go buy parts and assemble computers or build different things based on computers or use mostly computers as an interface to hardware. And I had done a little bit of work with the PIC microcontrollers, but I wasn't doing a lot of electronics. So that's one of the reasons why I stopped working in more web-based software, and I wanted to go back to actually building something.

Hodge: So this was 2002?

Banzi: Yes, it was between 2001 and 2002. I started at Ivrea in the spring of 2002.

Hodge: What were your initial projects like before you had the Arduino platform?

Banzi: When I joined the school, there was a teacher there that had just left, that was teaching with this thing called the BASIC Stamp, which is actually coming from a company here in California. And essentially, I think that BASIC Stamp was very important in the history of teaching tools for microcontrollers because they put a PIC microcontroller on a small module that had most of the parts you needed to get started and a way that you could reprogram the chip easily through a serial port.

And I had a fairly simple piece of software that ran on a Windows PC, so you would just wire up a few wires. It had a very, very, very simple Basic language. But it's something that you could teach in a matter of hours, and people could build small projects. So I started using that tool, and I was helping students build a little bit of-- I think one of the very, very first projects I worked on, there was a student who built this prototype of a t-shirt that could send a remote hug, so the idea that the t-shirt would hug you. The idea was they had a lot of pockets that could inflate with air, so we worked a little bit on that at the beginning. It was two students, an Italian and a Japanese student.

So when I started working on the BASIC Stamp, I noticed there was a lot of value in this system, which was much simpler because the other-- back then, a lot of people still programmed PIC chips using assembly language, which is really, really hard to explain to a beginner. It's way too low level. So the Basic in the BASIC Stamp was very few instructions. Also there was a lot of teaching materials, a lot of books that you can also just download as PDFs with projects you could just go through.

So that, I think, there was a lot of value in what they did. But it was very expensive in Italy because it was about $50 in the US, but by the time it was exported to Italy, added the duties, everything, it came to more

than $100. And already, it was a bit of a problem because you would have a small number of these. You would hand them out to students, and then you would have to collect them because they were expensive items. You didn't just give them away. There was an initial desire at some point to really move away from that and allow students to just build a lot of prototypes.

Hodge: What made you go down the path of doing your own board, or did it even start as your own board?

Banzi: Well, it's interesting because it wasn't required by my job. But I still felt that if we were trying to teach a new type of design, we were trying to experiment with a new type of learning, we should try to get a better tool, build better tools. So at the beginning, I started to explore a number of platforms that were available, so one of the people that was more well-known in our field was Tom Igoe, who later became the co-founder of Arduino.

Tom Igoe was teaching at the ITP, and in 2002, he was already putting all the information about his class online, which this is before there were blogging platforms. So he was just putting static HTML files on a server, but there were all these tutorials, all the explanation, the syllabus from his class. So he was really inspiring a lot of teachers around the world, and he was using another platform called the BasicX, which is another microcontroller platform, which is a little bit more powerful. So we started to adopt that one.

But there are a lot of issues. They would break down a lot. Students would either damage them or they would bring them to a state that would not be re-programmable at all. So we thought, we have to move to something that's cheaper, easier, so I started to experiment with PIC chips because this is something that you would find easily in Italy, for a reason which is kind of-- we can probably say that now, but back in those days, people were using PIC chips to build these cards that you would use to hack satellite receivers to get free satellite TV.

So I didn't do any of that, but when you went to these radio amateur fairs, there would always be a group of people selling PIC chips, PIC programmers, and a lot of other tools that would be very, very low cost because they were designed for you to hack your satellite receiver. But I just liked it because it was super low cost and available to buy. So I started to work on PIC chips. I acquired a copy of PIC BASIC, which was a little bit more sophisticated and had a better development environment.

I started to work with a couple of students on experimenting what it meant to create our own platform. So I remember I was working with a student called Lydia Soonesun. She was still building her thesis, so we started to build boards with just the PIC ship-- that was only a few dollars --and a PC to program them. And I told her, you're my guinea pig because I'm experimenting, trying to understand how do we create a better tool. That's the context.

Hodge: So at what point did you move away from the PIC chips and over to the Atmel-based platform that you probably became most famous for?

Banzi: So a number of these experiments I did led to the creation of a platform called Programma 2003, which was essentially the sum of all this experimentation I did on the PIC chips. So I went looking for an open source programming language for the PIC. GCC wasn't available. There were not C compilers available, so I used the compiler for a language called JAL, which was a very, very basic, simple language.

It was a little bit like Pascal, but it was open source, so you could compile it on all the platforms. Because one of the problems is that a lot of my students were on Mac, and a lot of these microcontroller tools are all for Windows. Maybe some of them would be for Linux, but not so easy to use. And a lot of Macs, for example, have USB plugs. They don't have serial ports anymore, and a lot of development tools still have a bunch of serial ports, which is really interesting how engineers are still in love with the serial port.

So I put together the experience I did with PIC chips, the work I did with the students. I brought in this JAL language, and I wrote the very, very, very basic development environment in Java. And this Programma 2003 came-- the name came as an inspiration was the Programma 101 from Olivetti because, again, I had recently read the story of the person who invented this computer.

Hodge: What was that computer?

Banzi: So it's called Programma 101, and it's considered the first desktop PC in history, at least this is also what Wikipedia says. And it was built in 1964 by a group of engineers at Olivetti. You could buy it in 1964 for $8,000, which was incredibly cheap for those standards. It had a very, very, very simple programming assembly type language. And it would be a calculator and also a programmable thing.

I was reading the story of that, so I thought, OK, let's call it Programma 2003 because I did it in 2003. And I started to use it with my students. And after I used it with my students, and I went through a whole year of teaching, I saw the value of having a cross-platform, simple-to-use tools, where the board itself would be very cheap to make because I just made the PCBs. They were only a few euros each.

And then I got all the students in a room one night, and we soldered 50 of them by hand, and we had our own platform, which was easier to use, more robust. They would make less errors because they already had a board with all the parts they needed. And if they blew up the chip, no problem, let's just pop it out, put a new one in.

So Programma gave me the overall structure of, OK, this is what you need. You need a board built like this. You need a programming language. You need an IDE. The IDE sucked, to use a technical term. But one of my colleagues at Ivrea was Casey Reas, who was coming from MIT, and they just worked on this programming language called Processing, which was a Java-based programming language that would make it easier for artists and designers to write software that would run on a PC.

So we discussed about the fact that we should really use processing to program this. And one of my students, Hernando Barragan decided to work on a master thesis to actually bring essentially the Processing language and user experience to hardware. And at some point, Bill Verplank, again, told me, well, you know, it's Stanford. Pascal Stang is building all these boards using these AVR chips. You guys should really look into that because it's running GCC. There's like an open source, functional GCC for this processor.

And so initially, the student wanted to build some kind of a Java virtual machine, and we said, no, that's not the way to do it. And then I think, it was also again Casey and Bill said, if you take C++ and you change it a little bit, you can make it look like Java. So you can make the language look like Processing but running on a microcontroller. So that's how then this student Hernando started to work on this project. That was the first project we did with AVR chips.

Hodge: How difficult was it initially to support something that was completely cross-platform for Linux, for Windows, and for Mac?

Banzi: Well, I have to say that most of my students were running on Mac, and I was using Mac, so I was concentrating a little bit on that. And also, I have to say that back then, for example, Macs used to come with Java pre-installed. Now, it's not pre-installed anymore. There's a number of issues around that, but it came with Java pre-installed. A lot of Windows machines had it.

So it was not that difficult because we used Java, which, for making the IDE, was not that difficult. And programming for multiple platforms, again, once you have GCC and a few files, you could do it. There were ways that you could generate makefiles for programming the microcontrollers. I think in that particular context, Java does a good job, or did a good job.

Hodge: So was it Processing [which] inspired Wiring, which inspired the Arduino IDE?

Banzi: Yeah. The general idea is that when I built Programma 2003, I tried to bring in a number of these elements from Processing, but the Processing source code was a bit impenetrable for me to be able to actually rip out Java and input this JAL language. So I just built a very, very basic JAL IDE, which was kind of inspired in a way to processing. And then when Hernando started to work on it, we said, oh no, you really need to start from Processing.

So we did quite a lot of work to isolate the part of the code that was doing Java, rip it out, and replace it with Java code that would compile the C++ code to make it then download into the processor.

Hodge: You'd mentioned this was an Olivetti-- had sponsored the school, was you were in one of their old buildings?

Banzi: Yeah, the building was called the Blue House because it was a 1956 building designed by this architect called Eduardo Victoria who-- the building won some prize back in 1956 for new architecture. And it was all covered with blue tiles. And it was a really beautiful building. One thing to say is that Mr. Olivetti, the original Adriano Olivetti, he was into design big time. Olivetti was one of the first companies to really apply design across the board, in the products, the graphic. The posters of Olivetti are collector items. The buildings themselves are all designed by famous architects of that period.

So this building was really nice, full of light, and it used to be the R&D office for Olivetti in the '60s, '70s, I think until the '80s. It was actually redesigned a little bit by Ettore Sottsass, which, again, was a famous Italian designer. Well, actually, he's famous for having designed also the Elea 9000, which is the first computer that Olivetti actually made in the '50s. And he designed the user interface in a way that is part of the user interface, so he was already used to working with technology because of the Olivetti thing, but he was invited to refurbish the building.

So the building was actually beautiful inside and outside, and it was very inspiring because you're near the mountains. There were lots of windows, lots of patios in a way-- how do you say? There's lots of like large balconies where you can go outside. And also, I would say, it was one of the very first buildings I worked in where there was Wi-Fi across the board. So you could really sit outside in the sun with the mountains and do your work. So it was incredibly inspiring as a building.

Hodge: Was there a cache of parts left over from Olivetti that you were using?

Banzi: So the interesting thing is that Ivrea the city is like 45 minutes to an hour to get to the nearest big, big city, which is Torino. See, if you needed parts, you couldn't really walk to a store. There was a tiny store with two old people working there. They would work with you only if they liked you, so it was kind of a strange thing.

So I needed parts. Going to these radio amateur fairs, I remember that I saw this company that was coming from Ivrea, so I went looking for them. And I realize they had basically a scrapyard. They would collect a lot of old Olivetti stuff. They had tons and tons of stuff piled up in this warehouse that you could just sometimes buy even by weight.

I remember I needed a bunch of cables to connect several motors to the Arduinos, and they had a full box of all the cables that old PCs were used to connect the speaker to the motherboard. And I said, how much is it for this box? There must have been thousands of those. They said, I don't know, let's see. They kind of gauged the weight and said, what about 20 euros? OK.

And then, they had boxes and boxes of LCD screens, text LCD screens that would go on printers. They would have lots of stepper motors because they would take apart all the different printers and the floppy

drives. So a lot of the projects we did for many years in Ivrea were based on recycling or upcycling, as some people say now, old Olivetti stuff. I think I filled up my car with keyboards, computer keyboards, and I think I paid them like the equivalent of like $1 each.

Then we took them apart, and then the students used the keyboard chip to learn how to do basic interactions. Actually, it was very simple. You could just use a cable to touch two contacts on the keyboard chip, and you could build small projects without knowing anything about electronics. But then somebody figured out that you could also use the-- there's like a Mylar field with all the contacts for the keyboard. Somebody built a gigantic touch-sensitive surface just by assembling a bunch of these. We would have these big boxes at the end of the lab where people would just throw in old parts. And so that's how we built a lot of stuff.

Hodge: Did any of these scrap parts eventually become part of the Arduino board itself? Did it influence any of the design decisions of the board?

Banzi: Well, I think when I was doing my experimentation with PICs, I'm sure I used some DB9 serial connectors that came from one of those Olivetti boxes. And the number of kits I built for the students for learning contained LCD screens from Olivetti printers that we sort of teach people how to connect to the Arduinos. Some parts became based off these kits I was building when we did classes. But they became part of mostly the projects the students did.

Also, after a while, students learned that if you see a big motherboard that has a lot of interesting parts, you could take one of these very hot air guns, and you can just hot air gun the thing and just shake it, and all the parts would come off. They saw me do that late at night one night, and they learned about it. So I said, OK, you can do it, but you have to wear gloves and use one of these pliers to be safe. So a lot of parts from these old computers became parts in projects.

Hodge: Interesting. What, then, caused you to move from the board you were developing around that particular microcontroller to finally go all the way and do what became Arduino?

Banzi: So what happened is that-- so I worked with Hernando as a supervisor and Casey-- this was the other supervisor on this thesis. So I guess there was a lot of work that went into that project, and for us it was important to have processing on hardware as a teaching tool because we were teaching Processing as a way to teach students how to program.

And I didn't want students to have to spend a month learning, I don't know, a random language and then a month after, to start again when they did work on hardware on another language. So having this uniformity was very important. But also, Hernando started off with this board that was using a fairly expensive chip. It was using this thing called ATmega128, which had 128 kilobytes of memory. It was only available as an SMD part and cost probably about $12, $15, if you bought it in small quantities.

And my theory was, it's just too much. It's too expensive. We need something much, much simpler that's through-hole so you can put it on a breadboard. So I think there was that kind of quote unquote, "tension" during the project because I had this idea of making something smaller and cheaper. So I started to look at the ATmega8, and I built a number of projects with the school using these smaller through-hole parts.

And then I started to build the different prototypes of my old PIC boards, but with AVR, ATmega8s. So the transition was that, obviously, I got the school to pay to manufacture a number of these Wiring boards, and I used them to teach my students so we've validated some of the features of the framework. It was much more powerful in a way and much more easy to understand, and a bunch of other platforms are there.

But also, on the other hand, I think there was a number of misunderstandings during that time because I guess the school was trying to get a bunch of students to sign forms that would transfer the ownership of

whatever they built to the school, which is normal in a bunch of schools, but some students reacted not very well. And I think Hernando, if I remember correctly, was one of those who didn't like this and never really wanted to sign this.

So there was a number of misunderstandings that made it for us a little bit more difficult to work on the project and take it forward. But for us, as a research project, we wanted to take it forward. I wanted to use it to build the new platform. And in a way, I think a lot of different kinds of misunderstandings led to the fact that, I think at some point when Hernando graduated, he left with most of the information about the board, apart from the executable themselves and a few boards. Since as a teacher, I signed to have them manufactured by a company in Ivrea, so I had at least the files to make the boards and everything else.

So I realized we had to redevelop something that would be open source from the ground up, so that we could continue working on that. So I think that the transition was, how do we make this into a platform that we can continue using which becomes more available to more people. And in the meantime, one of the considered co-founders of Arduino, David Cuartielles, came to the school to work as a researcher to do a project, which was based on extending Wiring with different modules.

So we started to talk. We said, it would be really cool if you had something cheaper and simpler that would cost a lot less than this. And we open sourced. So we would want it to be open source so that we can get a lot of people to contribute, so that was kind of the starting point.

Hodge: So this was 2004?

Banzi: This was end 2003. It was at the beginning of 2004, if I remember correctly. Because I think David was invited to work on this at some point in 2004.

Hodge: So open source software at the time had a lot of mindshare, but I don't think many people were talking about open source hardware back then.

Banzi: Actually, I wasn't even aware of the fact that there was a formal open source hardware concept. I only learned later that there was this work that, again, some American ham radio operators did. They did this TAPR license, which was kind of one of the first, or maybe the first, open source hardware license. I wasn't aware.

I had been a Linux user for many years, and I knew the power of having something which is open source, where you can involve a lot of people and get a lot of people to work on. I think, again, one of the misunderstandings between me and Hernando in the process we worked on Wiring is the fact that as a designer, I think he was much more, I don't know-- considered the product his baby, so I don't think he really liked the idea of having lots of people messing with his idea.

While for me, coming from a more software background, a lot of open source and Linux background, I really wanted this to be a platform that people could participate in, could just help out. And then, when we designed the first Arduino boards, I thought, why don't we just put them online? We just put the Eagle file, the CAD file-- the CAD software we used to call Eagle. Just put the Eagle files online, and let's just use them and see if somebody wants to contribute, wants to help out.

Also, I wanted to give a way to students, if they wanted to, to start off from an Arduino and build their own board. One of the things that the school did quite a lot is that the school had a fairly active exhibition design group led by a teacher called Stefano Mirti, which actually got the school to exhibit in a lot of different places. So one of the first things I did when I went to the school, I found myself working on an exhibition for the Biennale in Venice.

So we went in Venice, and we assembled an interactive installation. So some of the things that happened is that maybe a student would do a project which would work as a prototype, and then later on, we would have an exhibition so there would be a little bit of budget to take those projects to the next level. So we always did this thing of taking an existing circuit, maybe design it as a PCB, have the PCB made, and then build maybe two or three items that would be robust enough to be an exhibition. So I think that, again, was a driver for the open source software thing.

Hodge: It's interesting that you mention making these custom PCBs. I'm sure that was somewhat expensive or a pain back then, and it's still, 10 years later, somewhat expensive and a pain. Is that something you see changing any time in the near future?

Banzi: I think when I started working on Arduino, there was a lot of strange things that happened all at the same time right in that moment. From my experience of making PCBs when I was a kid or a teenager was a nightmare. You had to etch them by hand, drill them by hand. It was a really painful process. Also, you had these thing you to kind of iron with using your-- there was weird techniques to get these PCBs made.

And then, while I was trying to figure out, I found a PCB manufacturing company in Ivrea because there was a lot of electronics manufacturers, and there were companies there. And I found one which I started to use, and they liked the idea of working with a school that had all these crazy ideas, so they would give us reasonable prices and reasonable lead times. And the other thing that I think was useful is that there was a company that was not that far away from the school.

The first company I can think of in Europe that would you just send them a file and they would send a PCB. And they would do it even in 24 hours. Obviously, it would be like a PCB with no solder paint on it, just the fiberglass and the tracks, if you wanted. Depending on how many days of processing allowed, the price would go down. So I had a number of PCBs made like that because you could just go online, give them a file, and three days later, or two days later, this package would show up with PCBs you could just use.

So I think that was an interesting thing. And again, that company started off by selling the parts you needed to hack the satellites. And when they started to crack down a little bit on that practice, they realized they needed to find another business model, and said, you know what? We'll just make PCBs. And they started to make this online PCB thing. So that was the germ of something that now is really becoming a completely different industry.

Now you have people like Seeed Studio in Shenzen, where you send them a file, and they send you back not only the PCB, but an assembled circuit, working. See, if you design it with Eagle, and you use their part library, they have machines that are pre-loaded with parts, and they just-- boom. In a few days, for a very reasonable amount of money, they send a working circuit. So we are evolving, I think, to a moment where people just send a file and get back as many devices as they want.

Hodge: It's interesting that you keep coming back to the satellite hacking because a lot of US-- a lot might be a strong word, but certainly [part of the] history of US computing was phone network hacking in the-- I don't know if the '60s, but definitely the '70s. And so, it seems interesting how these large networks beget the next generation of hackers. At what point did you have a board that was-- you're ready to call it the first Arduino, or the zeroth Arduino, depending on how you want to count.

Banzi: In 2004, we had an exhibition that was something in the region-- I think it was October of 2004. We had an exhibition in Torino, and we had to fix a few of these projects to make them manufacturable. So we started to migrate a lot of these projects from wiring to the ATmega8. And one of my-- well, he was a student back then-- David Mellis, was part of that project.

And I said, David, could you give us a hand and try to figure out if you can take Hernando's code and re-implement it as an open source piece of code? And that was like one evening at like 11 o'clock at night,

something like that. And then, as a bit of a provocation, I think I said something like, somebody smart like you can do it in probably, what, in a week or something? Which is the best way to trigger-- he was an MIT graduate, really, really good software developer. So he was like, hmm. Doesn't speak very much. So he sat down, and like two hours later, he said, OK, I got this.

And he had like the first germ of the first Arduino because essentially, it took the commands that Hernando imagined for Wiring and imported them to a piece of software he built from scratch, in a way, to run on the little ATmega8. So we took an existing program we did on wiring and move it over to this platform. And in that particular moment is when I got a PCB made which was essentially like my Programma 2003, but with the ATmega8 on it.

And there is a picture online on my Flickr where there is that board. There is a parallel port connector badly soldered on the PCB, and then there is a LED and a resistor, again, badly soldered in one point of the PCB, which was the first time that I got that port to blink an LED with David's code, which you had to compile with a makefile. There was no IDE back then.

So that's when we started to transition towards data. And with David Cuartielles, we started to do a lot of work. Let's figure out how to turn this into something a little bit more. So at some point, it started to be called wiring light. So the board took different shapes. And then, at some point, I would say at the beginning of 2005, we had this idea that we have to start using this to teach. So we have to prepare for this thing.

So we hacked the Wiring IDE. It's written in Java, so you can reverse engineer, change a little bit of things. So we hacked it so that instead of compiling Wiring, which compiled Arduino, and then we made these bonds, and then we said, OK, we need a few boards to run a class. The first workshop was supposed to be at the end of March of 2005 in Malmo, in Sweden where David teaches.

So he took the design of the Arduino board and made it the shape it is today, which I remember having a conversation-- why do we have this weird shape at the end? And he was like, just because. Because then it becomes different. It's not rectangular. OK, it's true. Aren't you tired of this rectangular thing? The size wise comes from the fact that I wanted it to feed into this-- I had found these nice business card plastic boxes, and I wanted a PCB to fit in there as a project box. So the size comes from that.

Then David gave it the shape. Then we had a lot of meetings with this company near Ivrea doing print circuit boards called System Elettronica. And there is a person that runs that company called Mr. Apruzzes, who is an interesting character. He's somebody who lives-- his life is printed circuit boards, so he's like a massive printed circuit board geek. And he's kind of interesting, very energetic, very flamboyant, and who likes to talk a lot.

When we went there to meet him, we always had long conversations about PCBs, so he started to say, you should really make your PCB blue because somebody determined that blue color is easier on the eye of the people who have to assemble circuits. So some of the features of Arduino sort of happened like that.

Hodge: Was it related to the Blue House at all?

Banzi: No. No, it's just this person said, somebody did a study and determined that blue is easier than green on the eyes of people who assemble the circuits. So we thought it was very responsible for us to build something that would be easier to look at. And also, again, there was this underlying desire of having something which was not the usual green rectangle, in order to be distinctive. You're in a design school, and you make a green rectangle like everybody else.

And so what happened is that, we had the board, we had the shape, but we didn't have a name. One day, I think I dig all throughout my emails, and I determined it was the 17 of March, 2005. I was sitting in front

of the computer with the CAD file. Mr. Apruzzese is on the phone, saying, look, either you send me the file by 12:00, or we're not going to be able to make the PCBs for the end of the month when you need them. So, David, what the hell do we do?

So, I said, you know what, let's call it Arduino like the bar. I used to go to this bar in Ivrea called Bar Di Re Arduino to get drinks. So I thought, let's call it Arduino like the bar. Then we'll find another name later on, but let's start with this. So we put the name in, send the file to this gentlemen who made the first-- I think we made about 300 of them, where we had about 150 serial and 150 USB, which had a terrible mistake I made on the USB thing. I swapped the D-plus and D-minus. You couldn't really solder the connector on it, but still, that was the first-- sorry?

Hodge: Were you able to make them work?

Banzi: You could solder a cable. You could take a USB cable, chop off the connector, and solder wires directly. That was a way. And there are some pictures online of people doing that and then using a zip tie to hold onto the cable. But then, what happened is that, more or less during the same time, around 2004, I guess, we started working with a local engineer that I found at one of the PCB companies. There was this guy working there doing microcontroller development or whatever.

So I started to collaborate with this guy, Gianluca Martino, and then, we started to hire him to come to the school to build some circuits, build some help, manufacture things, or he would just find manufacturers to build stuff for us. So I think Gianluca designed this tiny PCB that would basically rearrange the pins on the connectors so that you could salvage some of those boards. It was like a tiny PCB where you would mount to connector and then you would solder this onto the Arduino boards, but we didn't salvage many of those.

Hodge: So one question I do have to ask, since we're talking about the PCBs, is probably the most infamous design quirk, we'll call it, of the Arduinos is, I believe the space between the headers on the top is 0.16 inches instead of 0.1, which has caused many hobbyists much hand-wringing. What's the story behind that?

Banzi: Well, the problem is, when I was building the first layout for Arduino, I made a mistake. I didn't align the connector properly. So every other connector is aligned with the right spacing, and there's one that's kind of half of that spacing. And I didn't catch that, so we replicated that on the USB thing. And then, we started to try to use this-- how do you call them in English? These kind of proto-boards, perfboards, in a way, to build modules that go on top of the Arduino.

We realized that the alignment was off and it's-- oh, what a nightmare. And it's weird because when we started the manufacturing the first, I don't know, 1,000, 2,000 Arduinos, for us, the idea of telling 1,000 people that the shields they got-- shields are the modules you put on top of Arduino. We came up with that name as a way to have something that wasn't technical, that was funny. So the idea was that Arduino originally was a king, so that's the king's shield.

Now, there's like billion dollar companies who are using that name that don't even know what it means. But if you go back, some of the names inside Arduino have these weird stories. So the shields-- we thought, we cannot not tell people they have to throw away all the shields. Now there are millions of Arduinos around. And there's a bunch of, again, billion-dollar companies that have adopted that layout. We thought, OK, what can I do? I can't fix it now. This is it. It becomes a feature. Somebody feels that it's useful because it forces you to plug the shield in the right direction because it's got this gap.

Hodge: It's a good problem to have when your product becomes so successful that you have to maintain backwards compatibility with things like that.

Banzi: Yeah, in 2006 when Arduino reached 10,000 units, I remember that Phil Torrone of Make Magazine wrote an article that was, I don't know, it was an incredibly enthusiastic article. It felt almost like when Apple introduced the Mac. Oh my God, 10,000 Arduinos. It felt like a lot of success.

Hodge: Did you initially plan for the shields? Was that part of the initial design?

Banzi: Yes, because one of the projects that-- so David Cuartielles was invited to Ivrea to work on a project related to extending Wiring. And the main objective was to build what-- at the moment, they were not called shields. But the idea that somebody asked me, if there were some money to put in Wiring to extend Wiring, to do something on Wiring, what would you do? And I said, we could develop a bunch of modules that go on top of the board that extend the functionality of the board.

So we decided to develop a-- I think there was a motor shield or something that would have a chip that would control stepper motors, DC motors, so people could build robots or similar things. And maybe, I think, David worked even on a Bluetooth thing. So there was two or three of these shields that were part of the project that David was doing.

And then, in the end, they became useful because then when I ran one of my classes, one of my students used that motor shield and the wiring board to build a haptic interface to control motor and a knob to build an interface. So that came from that. So, obviously, when we did Arduino, we thought, OK, we need to translate to Arduino the work we've done on the wiring modules, and they became shields.

Hodge: So picking back up, we have the first Arduino out. We're starting to see the first shields. At what point did you develop into not just a school project but actually a company, and what was your idea for a business model?

Banzi: So there is an interesting fact that happened around that time, is that we learned that Telecom Italia wanted to shut down the school. At the moment, I thought, oh, wow, this is going to be a problem because if they shut down the school, and somebody comes and says, OK, everything you did at the school is now property of Telecom Italia, because Telecom Italia at that point was the incumbent telecom operator in Italy, and they owned Olivetti. So they owned the school as well.

So we thought, maybe some lawyer shows up and takes all the projects and says, these are ours, puts them in a drawer and then we forget about it. Luckily, we already had open source software, open source hardware, even though we didn't really know. There was not a main reason. But that was a big push, again, to formalize that. So we had a situation. We got a little bit afraid that the project might be stopped in some way.

So we pushed on the open source hardware aspect, and also our software was already open. But then, our problem is that we needed to manufacturer this board. We didn't want to assemble them by hand. So as I mentioned, we had done some work in the past with this person Gianluca Martino, who was a local engineer that was doing local consulting work and helping out. So we hired him already at the school a number of times to help out, build some stuff in projects.

And so we asked him if we could help out, manufacture some of these devices because we thought, if we can maybe make 200 of them, I'm pretty sure we can sell like 50 to my school and 50 to the university in Malmo, where David was. And the other 100 I'm sure we can probably sell them out to people. I had given away a bunch of those blank PCBs I made originally. I gave them to people as a way to seed the community, and a number of them were actually able to assemble the Arduinos.

And some of them even came back and said, can I buy some PCBs from you? So yes, so we just said, give me five euros, I'll give you a bunch of PCBs. But this was not something you could really do. We had to have pre-assembled boards, so we basically asked Gianluca to do it. And so we started this process of

essentially-- OK, the board cost x, we sold y. This is what's left. What do we do? And we said, take that money and use it to build another batch of boards.

Because I think one of the biggest things that happened is that-- also very relevant to the history of Arduino is the fact that when I created the Programma 2003, one of the big limitations is that it was only me. And it's very hard to start something like this completely by yourself. So you need to have a team around you of people that can share the excitement, help out.

So that's why, for example, in Arduino, I started to work with David, and then there was David Mellis, so we took him on board, started to collaborate with Gianluca. And then, a while ago, I had met this Tom Igoe that I mentioned before that was teaching at ITP. I said, Tom, you need to come to Europe. We need to meet up again. I met him briefly once in the US. We need to work together on this thing, so we showed him. He liked it.

So I think one of the defining moments is when Tom decided, OK, I'm going to bring back these boards to the ITP in New York so that we can work with it because he was using, again, a bunch of products he wasn't happy with. So that moment where the board travels from Ivrea to ITP, it's when I think that Arduino had the chance to become something relevant.

Because before it traveled to the US, we were going to just make a few of them, but it was never going to come out of the shell. And so I think when they started using that board at the ITP in New York is when really Arduino started to shift from being something that we used internally into something that had a community.

Hodge: What was the idea behind Arduino as a company, given that you were giving away your hardware and giving away your software if other people wanted to make it? How were you going to be different and continue to have people pay you?

Banzi: Well, at the very beginning, we didn't have a problem with people cloning Arduino because there wasn't a market. People didn't really care. And most of the people we worked with-- first, they didn't even know how to make a PCB. So for them to be able to buy a PCB was already good, let alone assemble SMD circuits. So the fact that you could have an available and SMD assembled circuit, that you would just plug into a computer, and it would work, was the selling point.

So even if it was open hardware, at that time, it wasn't like now, that you can just find Seeed Studio in China. You send them a file, and they give you a working device. You needed people in the factories that would help you prepare for that. So in a way, I think at the beginning, the business model was basically based-- we are Arduino, we sell the boards, and that's it. But at the beginning, again, the business wasn't huge. There wasn't like millions of people ready waiting for us.

So it was a very, very long and slow and somehow painful process of spreading the word. I remember me and David spent a lot of time going around Europe. He went to India, even, went to Canada to run workshops and teaching classes and doing presentations where lots of times, we didn't even get paid by the people who organized the workshop. We just went there just to be able to say, we have this.

So we paid for our own trips, or we paid for our own-- or maybe I was teaching in a school somewhere, and I would just give a presentation, give a workshop. So that was a very painful process of leading dissemination of getting people excited about this. Then later on, there were some people starting to make boards that were compatible in some way. So the problem was, how do we organize a business around this?

So the original idea was, well, we have a brand. We try to protect the brand in a way so that people-- we decide what gets to be called Arduino. Anybody else can build it, but they have to call it with a different name. I think it came out of long discussions going over a number of months of figuring out.

Hodge: So when you talked about moving the board, traveling across the Atlantic to the US, somewhere around this time period, O'Reilly came out with Make magazine as sort of targeted for a new generation of do-it-yourselfers. And over time, Arduino became incredibly synonymous with this. How did that happen? How did you first meet Arduino? I mean, excuse me, how did you first meet O'Reilly, and where did you discover that you had these things in common?

Banzi: So luckily, Tom knew O'Reilly, the company, already, and I think Tom and then O'Sullivan wrote the book called Physical Computing, which is actually the name that we give to our discipline in school, physical computing. So they wrote the book for another editor, and they were thinking about maybe making other books or something. So I think Tom knew the people from Make.

And then when Make started to look for people who could write articles, and Tom was one of the people that was involved as a author, but also I think what happened is that Make was also looking for a platform to give to their audience so they could build things. So actually, they went out and got a company here in California, I think, to actually build this thing they called the Make controller. And they built a group of people that were advisers to that project.

And I remember that when I read it on Make Magazine, I got incredibly pissed off. I was like, what the hell? This thing is massive, expensive, complicated, and the compiler is like $3,000. Who the hell is going to use this thing? Obviously, I thought that Arduino was better. At least it was cheaper. It was like, 20 euros, so $29.

So I remember probably writing to Tom saying, what the hell is going on here? You are on the advising thing, and they are doing this. Make is trying to create a platform. Then, what happened is that Arduino started to be used by students at ITP. They made beautiful projects. These projects would be displayed in their end-of-year show. People would go to see these projects, see them online. They would see them at the show, and they said, oh nice, what is this? Oh, this is Arduino. OK, that's great. Oh, that's interesting.

So the word kind of spread out. And some of the first projects based on Arduino started to show up on Make Magazine. And then, the community made their decision. They thought that the Make controller was probably a little bit too complex and high-end for what they needed. And this 8-bit microcontroller with a super basic language was probably more suitable for them. And they just adopted it.

Hodge: What year was that when you really saw that adoption uptick?

Banzi: Probably the beginning of 2006, something like that.

Hodge: And what was the maker community like then? What was the demographics? Young, old, students?

Banzi: For me, it was kind of interesting because there was-- for me, back then, the community was different. The "maker movement," quote unquote, as you understand it now, was probably happening in the US, and I wasn't coming to the US a lot. I didn't even have the money to do that. Because for a long time, there was basically not really much money to be made on Arduino. So I was still teaching, and I was running workshops, and I was doing consultancy projects.

But in Europe, a lot of the places that we used to hang out were the places where there would be designers, artists, musicians. For example, I saw a bunch of people that were interested in music that got a lot of Arduinos, built software that you could use to interface sensors and bring them into a computer and then use Maximus P or PD, different software you can use to build sound applications. And they would experiment and build their own musical instruments, people who built installations.

So our community in Europe was more the designers, artists, musicians, weirdos, people doing that kind of stuff. But I guess some of the projects they were building, they were more interesting than what you would normally find in an electronics magazine because the problem, I think, for the old electronics magazines is that the projects were usually quite, sometimes, abstract, too much in love with the technology itself and less interesting what actually I'm building. There's only so many radios you can build in life.

So I guess that what was interesting about the maker movement is that it was a bunch of people that were not coming from the technical, technology field. They were actually interested in building things that meant something for their life. And they actually didn't care what technology they were using. So a lot of the people who are into technology, they can debate for days if it's better to write something in Python or in Ruby or in Java.

A lot of the makers, they didn't care. They wanted to build a fantastic interactive installation, and if you told them that you could do it by writing it in, I don't know, assembly language and that was the best thing, they would just do it. If you told them that you could use Microsoft GW BASIC on a DOS machine, they would have done it. Because they wanted to build the fantastic installation. They didn't care about the technology. I think that was one of the interesting things that a lot of these non-technical people brought to the maker movement, an attention and interest in the final product, and less obsessive attachment to the right technology.

Hodge: So did the uptick by the maker movement in the US, did that drive your design of future boards? Because you've released a lot of boards since the original one. Were there any boards in particular that were influenced by the way people were using them?

Banzi: In a way, the fact that we also run a lot of-- and still do run a lot of --workshops is also because you sit down with people, and you watch them build. And then you're like, OK, I think what people need now is this or that. So a lot of the design of Arduino boards either comes from the projects that I was doing, or we were doing, when we were doing a lot of consulting.

And a lot of my consulting back in those days was exhibition design, so doing interactive exhibitions, or watching people and seeing what they wanted. So for example, one of the early variations of Arduino we made was the Bluetooth board. And the Bluetooth board comes from working with a bunch of artists and people doing installations saying, I want an Arduino board that communicates wirelessly with my computer.

In fact, the Arduino Bluetooth was the first one that has a power supply on it that you can run off a battery that would last for a reasonable amount of time. That was a feature that was requested by the community. A bunch of people wanted that. Or while, for example, the Arduino Mega comes from the fact that Gianluca had a client that wanted to build a vending machine.

This is an interesting story because the vending machine was selling some pretty peculiar products. And the idea is that with an Arduino Uno, you couldn't run all the motors and everything and a reader, so we said, OK, we need to migrate Arduino to a larger processor, which became the ATmega1280, which had 128 kilobytes of flash and had a bunch of serial ports.

Hodge: Was that the one you had vetoed earlier in the design of the original Arduino?

Banzi: Yeah, it was a similar product. That was the ATmega128. This is the 1280. It was like a slightly more modern version of that. Well, originally because as an introductory product, lot of people-- it was too expensive, too complex. Then here comes the problem where some of the Arduino boards were too small, so people wanted more IOs, more memory, more whatever.

We built that PCB motherboard for the vending machine. I remember that I ported Arduino to that processor during a weekend in anger. I was upset for something else that was happening in my life, so I spent like a weekend. And it was a bit of a crazy-- I hammered Arduino to run on that. Then David did a much better job later on. But I had a version of Arduino running on that processor, and it was powering the vending machine.

So it then became the basic design for making the Arduino Mega. Then, the Arduino Ethernet comes from-- the Ethernet Shield --comes from the fact that I was doing a lot installations for exhibitions. And in a lot of situations, you needed network connectivity because you couldn't just run a USB cable for like seven meters. You needed something that you could use a cable to run data.

And I think one of the first projects where we had the problem of connectivity was this project where we had a long, long table with projectors projecting from the ceiling, and we needed to read a bunch of sensors, and the sensor would change the way the projection happened. So we needed to carry the data over, so that's when we started to say, OK, we need some kind of a network connectivity, so we started to work on the Ethernet Shield. So a lot of the things either came out of working on a project or observing people and understanding that they needed a certain tool.

Hodge: The maker movement looks like it's gotten younger over time. If you're in a Maker Faire, you see a lot of families there. Has there been anything you've done to outreach to kids in particular?

Banzi: I would say not specifically, but it just happened naturally that younger and younger people started to pick up Arduino and build things. The more Arduino became famous, the more books came out, the more tutorials, the more projects. And obviously, some parents started to use Arduino with their kids. So obviously, that created a little bit of a community of kids. I remember early on, I had a workshop in Amsterdam, and a woman showed up with her teenage, probably 13-year-old son, and they were learning Arduino together. So I noticed that it was more and more kids getting into that.

Hodge: Why is that there are wearables and smartphones more powerful than they've ever been before and more prevalent than ever before, so why are people interested in working with fairly weak processors and having to do all the work themselves?

Banzi: Well, one of the things I think I noticed is that when you work with kids, I guess they are born in a world where a high resolution touchscreen is kind of boring. Everybody's got one. If you're a kid, and you were born from the late '90s onwards. LCD screens and touch screens are part of your life. But then when you hook up an LED to a circuit, and suddenly, you can blink this LED, you can change the brightness, you can control that, I think you get a feeling that you have much a different level of controls over the world.

So I think, in a way, it becomes more exciting for some kids, this idea that you're not doing another screen, but you're doing something which feels more physical, more real. And obviously, for certain kids, then they can build robots. Building robots is something that people like.

Hodge: How prevalent have Arduinos become in robots?

Banzi: Well, there is a ton of kits for teaching robotics to children that are based on Arduino, programmable with Arduino, have an Arduino inside, so there's lots and lots of them. I could make a list of 20, probably, just off the top of my head. Again, because it's something where you can build a robot very cheaply, and it's sort of easy to program if you build it.

My co-founder, David Cuartielles, has actually got a grant, I think from-- I don't remember which country -- to go to Mexico and teach robotics to kids, but the robots had to cost less than $20. So he made this very simple PCB, used super basic motors, and then used a ping pong ball as the front wheel, and built a

small software library that would hide away some of the details of controlling the motors, and had 10-year-old kids buildling robots.

And then in the Mexican tradition of the fighter masks, they built some "masks," quote unquote, to build on top of the robots, and they had these kind of little carts moving around. These were underprivileged children, and they got into it big time. So in a way, those things are very nice to observe.

Hodge: Were those use cases you ever imagined with the Arduino, or did you think it was really going to stay with the more professional set, with artists and musicians?

Banzi: At the beginning, I was concentrated on building something that was interesting for my audience, which is interaction design students in master programs. Very, very, very narrow focus. So for me, the fact that our product could become successful in other schools was the cool thing.

But then, it escapes because a lot of the students, they take a one year master. They leave the school. And so, in a way, it was like a little virus. So a lot of the things that happened afterwards were sometimes unexpected, or we see it happen, and we try to jump on it, but in a way, a lot of people provided ideas that we didn't expect become relevant.

Hodge: Has that adoption in other areas changed your philosophy at all of the product you're delivering as to what your goal is?

Banzi: Well, in a way, we try to stay true to the original idea of making something which is completely open source and that people can really build upon. And we really want to make a tool to empower people. So I think those elements are still there. And obviously, for me, it's important to work with more and more complex technologies and make those available in the same philosophy as Arduino.

Obviously, as the time progresses, I can see also from my students that what was super cool for my students in 2003, which was connect the sensor to something building flash on a computer-- that was like incredibly cool. Now, we would consider it slightly lame. But then, a lot of people now are excited about connected products. So they're finding where they want ways to create connected products without the need to become a hardcore software engineer.

Hodge: On that note, your latest products have become a little more network-centric and that network functionality has been built into them. Is this a result of, for example, the Raspberry Pi hitting the market?

Banzi: No. We started to do connected products before there was a Raspberry Pi. So the Ethernet Shield came out in 2007, but it was something that was in gestation before the Bluetooth came in 2006. There were some experiments in making Wi-Fi stuff early on, but it was very expensive. In fact, we did have a Wi-Fi shield that came out in 2010 or '11, but it was very expensive.

But back then, the other Wi-Fi modules for Arduino were-- we're talking $89 for a Wi-Fi shield back then. Now, that price wouldn't be acceptable for the market. But back then, it was complicated to connect the device like an Arduino to a Wi-Fi network. We are following the fact that the community is more and more interested in connected products, so they want more and more ways to build these kind of devices. And I think they like the conceptual model of Arduino.

Because the Raspberry Pi is really cool, is a really cool project. But in a way, it's a computer with Linux. So if you're not somebody that can sit down in front of a Linux computer and write code, it can be a problem. While Arduino has a very defined programming model, it says there is one way to do things. That's it. Which is the opposite of what, normally in the technology world, people-- they tend to like when you have multiple options for doing the same thing.

When you're dealing with beginners, with makers, sometimes, they say, just tell me one way. I'll do it that way, that's it. So I think people like to work with Arduino because there is this very well-defined conceptual model, programming model. There is one way to do things.

Hodge: Is that getting harder to maintain when you have network connectivity? Because something like the [Arduino] Yun, if I understand correctly, you have your traditional micro, but you also have a small Linux computer on there, and you have to get those two to talk to each other. How has the adoption been of that, and how has it been teaching people to use that?

Banzi: That one still came out, but the fact that, at some point, you started to see on eBay a bunch of these little Wi-Fi routers from TP-LINK that you can even buy for $15. And people figured out you could put a different operating system on it, so they put this OpenWrt operating system. And then, they figured out that there was a serial port on those chips, so you could hook it up to an Arduino.

So people started to build these projects where they used these Wi-Fi routers to do Wi-Fi connectivity, but also, since they were running OpenWrt, you could write little scripts in Python or Lua to run on it. And then, they would use the Arduino for the I/O. So I was looking for ways to combine the two and try to make a product which would be the two of them because the idea was, OK, let's enable that. Let's have a product where you delegate the Wi-Fi connectivity and some of the more complex stuff to the Linux side, and then you have an Arduino.

But our model, it's not-- you program the Linux side and then the Arduino is a slave. But again, in order to fit in the Arduino model, you basically program the Arduino processor, and then we wrote a bunch of libraries that encapsulate a lot of the features available in the Linux side. So if you want to make a network connection, you use the same code you use on any other connected Arduino, but this goes over to the Linux side and tells Linux things to do.

And then, once we had that capability, we develop again a programming model because again, there's a bunch of boards you can buy around that do that. It was like a Linux thing and an Arduino thing attached. But nobody ever spends time trying to develop a programming model for them. They say, here's a piece of hardware. I made it cheap. Good luck. It appeals to a bunch of people that are experts in programming, but it is an issue when now you want to work with beginners.

Also, this kind of combination has an advantage that since you have a little Linux machine on it, you can run some code on it that allows you to build IoT [Internet of Things] applications, connected applications, without having to write the server component. Because again, a lot of people don't know how to write the backend service and put it on a server somewhere. So that little machine worked as a kind of a micro-cloud on board.

Hodge: So the people who found these-- did you say they were TP-LINK or D-Link routers-- and they put OpenWrt on them, and they had serial ports. Was any of that code ultimately contributed into the community and given back to the Arduino project?

Banzi: Well, those projects, I observed at the beginning, they were very basic hacks. Somebody wrote like a very simple Python thing. They would talk to the serial port to Arduino. They were hacks. So in the end, we took OpenWrt, and then, we wrote our own thing called the Bridge, which is a Python processor on the Linux side, plus a library on Arduino. And then, I would say that a number of people started to use that code on similar hardware.

So people took our libraries and started to run them on-- they went out and bought the $15 router, wired it up to Arduino, and re-used our code. Or there was a couple of other companies that actually built similar boards and used our code. Because again, when you build a product for the maker audience, just making a piece of hardware and putting in on the market doesn't do you very good.

You need to have-- what is the mental model? What is the overall experience they are trying to design into the product? And that's made of software, IDE software, libraries, a way to represent certain concepts that might be complicated, documentation and tutorials, and then hardware. So this Bridge library we built created a model where people could get their heads around this system, and they never needed to program the Linux side. So a lot of the people that used the Yun, they never touched the Linux. They don't even know that Linux is there. They just do everything in Arduino.

Hodge: To what extent that we've talked about the community that's sprung up to use Arduino helped-- what's the community like that sprung up to help contribute to Arduino development in terms of your open source repositories and things like that?

Banzi: So there is a very active community with a large number of developers who contribute a lot, and it's pretty amazing to see. But also, for example, there's a group of probably 20 people who are moderators on the forum who do an amazing job. And to me, their contribution is at the same level of somebody contributing software in the GitHub. Because they're spending hours every day on the forums, answering questions, helping people, getting rid of the trolls, which when you're building a community you need those kind of heroes in a way.

So you need hero developers that help you on the code, but you also need hero moderators. So the contribution for [INAUDIBLE] it comes from different areas. I would say that, looking back, the contribution on the hardware side has always been fairly weak because normally, when people imagine like a different Arduino board, they tended to just go ahead and build it themselves. So there were only a few cases when people said, OK, I found something you could fix in this Arduino board or whatever.

So they tend to build the board and say, oh, I made my own whatever Arduino compatible thing. On the software side, you have a lot of people that contributed a lot of powerful features to Arduino. And obviously, there are some star developers. There's this person called Paul Stoffregen-- I'm sorry, I don't remember how to pronounce the last name --who has a company that makes Arduino compatible boards. But he actually contributes a truckload of code to the Arduino project.

The only thing I would say is that there is always a little bit of a tension because the people who are able to contribute to the code in Arduino on the IDE and on the system software, they tend to be experts. So they tend to ask for features to be added to Arduino that would turn Arduino into the professional development tools we wanted to move away from. So there's always a tension of trying to explain, look, it's not that your code is not great. It's great code. We love it. It's just that it will make Arduino like the stuff we went away from. So we cannot accept this contribution. I'm sorry, but don't take it badly. Some people take it badly.

But in a way, there is this issue that we have to keep in mind, that Arduino has a certain philosophy, and we can accept only the contribution that feeds in that philosophy. And a bunch of people like Paul, they completely understand the philosophy, and they provide contribution that makes sense.

Hodge: Exactly. Arduino IDE currently is so minimalistic. It's almost difficult to see it as an IDE when you compare it to, quote unquote, "professional" IDEs that have menus and buttons everywhere, and I guess that's part of the appeal. You write the code, and then you push one button, it's uploaded, and things start blinking.

Banzi: Well, that, I think there was the, quote unquote, "the great innovation of Processing." You make an IDE which has six buttons, and that's it. The menus are not much more than, I don't know-- there's probably 25 menu items in total. I think if you're a beginner, and you are not really into technology to begin with, the visual complexity of something-- could be a board, could be an IDE, could be a website -- tends to scare that user, tends to say, this is too complicated for me.

So one of the examples I give is that, there it this professional development tool which is very powerful, called Eclipse. But if you put a beginner in front of Eclipse, it is a lot to take on. It's just way too much stuff. And then, when you look at Arduino, it's six buttons. So you're like, OK, six buttons, how hard could it be?

Hodge: Recently, you've had some fairly big announcements. You have had, or have, a partnership with Intel around the Galileo and the Edison. And then, I think what was an extremely surprising announcement to just about everyone is a partnership with Microsoft, who has not been traditionally known as a friend of open source, at least in the last decade or so. So would you like to talk about those?

Banzi: Well, for me, for example, as a techie person, I probably spent a good 20 something years hating Microsoft. But then, I guess that when I started to become older, I started to realize that in a way, the current Microsoft, the one that's happening in the last two years, is actually changing a lot the way they are. And they understand that open source is now an important part of the way people expect to develop software.

If there are no open source libraries and everything else, people are like, why would I do this? It's not open source. So they actually embrace that, and they open source a lot of code. So when we were approached by Microsoft to work together, obviously there was an initial, ugh. Like, hmm, OK. What are these people trying to do? So actually, we took a lot of time to talk to them, and it took months to do this. Because we talked to them.

We wanted to understand and meet the people, and then I met a bunch of people that I felt were genuinely excited about the maker movement. They really wanted to change Microsoft. They felt that finally, the top was telling them, go. Open up. And so, we liked the team. If we work with that team, and everything that we do is open source, so the idea for me is very important.

If you work on open source, and you are really an advocate of open source, it's too easy to hang out with your own people. It's like if you go to the Maker Faire, and you feel that everybody understands what you do, and you get this warm feeling in your heart. But you're not really, fully serving the cause of open source. Working as an open source advocate means, Microsoft wants to work with you, and they say, we want to do everything. The work we do is going to be open source. Then you stretch your hand, you say, yes, OK. Come over to this side, and we'll work together.

And I think that's important, and obviously, It's controversial. A number of people told me-- some people on Twitter told me I was a sellout and some other bad words. But it's too easy to hang out with your own people and say that the rest of the world doesn't understand you. You have to reach out, go talk to the people that have a different philosophy and are willing to do what you do and pull them in.

Otherwise, it's too easy. Life has become too easy. I hate everybody. I only hang out with the people that really think like me, and everybody else has to-- it's too fundamentalist. I don't like fundamentalist. I like to create bridges, and I think that's one of the reasons why I work on this. And also, I think it's about-- there are millions of developers who develop code for the Windows platform. That's their platform. They don't understand and they don't work on anything else.

A lot of them never had a lot to do with open source as well. When we were at this Build Conference in San Francisco a couple of weeks ago, there were people approaching the Arduino booth and saying, oh, wow, what is this? So if you go to Maker Faires, you are hanging out with people that, oh, yeah, Arduino. My six, seven-year-old son does Arduino all the time. It's too easy.

It's interesting to go to an event where people are like, what the hell is this? And they say, what kind of operating system does it run? And I was like, well, there's no operating system. It runs code. Oh, really? So this is very interesting for me to be able to take those developers and show them what we do in open source, show them about open source hardware, enable those people to do hardware.

And these are the developers who develop the software we use every day. You go to the cash machine, then you go to a restaurant, and the machine they use to book your table, it's all stuff running on Windows. So to me, it's important to help that community become part of our community.

Hodge: The most interesting thing about that announcement to me was that you were talking about, if I understood correctly, that sensors in phones could become first-class citizens to Arduinos, so sort of the idea we're already carrying all the sensors, let's make Arduinos talk to them.

Banzi: There are some interesting things are happening right now. If you go to a shop now, you can buy a smartphone, a Nokia smartphone, so something with a fairly high quality smartphone, for $49 including SIM card. And this thing has accelerometer sensors, connectivity, Wi-Fi, GSM, Bluetooth, the list is-- and it runs on a battery for days. If you don't use it too much, it can even run for a few days on a battery. And it has cameras and all of that.

So it is interesting the work that Microsoft did were to create the Bluetooth connection between the phone and an Arduino so you have a library for Arduino coders, which basically wraps all the functionality of the phone and makes them available to you like you had a bunch of shields piled up on your Arduino. So you can use it to connect to the network, do Bluetooth connectivity. You can even draw on the screen, a user interface, so you can use the phone as a UI for a project. So you can put buttons, and you touch them, and you get a message back that you can deal with.

So I think that it's potentially quite powerful as a combination. And on the other hand, they actually added to the classic programming languages you use on a Windows phone. They added a library that, again, through Bluetooth controls an Arduino. So in that case, if you're an Arduino developer, the phone becomes a platform for you. If you are a Windows developer, you can work with Arduino without having to bring all the development environment in. You can just stay in your tool.

I think that was pretty interesting because we're going to have-- Nokia, I think, announced this feature phone which has network connectivity for $29. And it runs for 30 days on a charge of battery. If we were able in the future to connect to this phone as well, I think phones are going to be the next maker platform.

Hodge: Your other major announcement, a little older but with a major company, was with Intel with the Galileo and the Edison. How is that fitting in with your vision?

Banzi: One of the original ideas I had about Arduino was the fact that there were a bunch of different products on the market where the hardware might be good, but the software was always kind of, eh. So I thought that Arduino could, at some point, become some kind of a Esperanto of microcontrollers, like a language that you can use on different platforms.

So the original idea for Arduino was to try to run it on different platforms. At the very beginning, I did an experiment of running Arduino on an 8051 microcontroller just to see how hard it was to port the code. So my idea was that you would learn how to use something like Arduino, and those API would follow you on different systems. It was obviously just the beginning of an idea. And then, when we started working with Intel, I thought, OK, this is the chance now to really push on that vision.

The Intel machines are Linux machines, and you're writing, essentially, a Linux application which is using the Arduino API. But you're building a command line Linux app, which then uses the operating system to connect to the hardware. So I think what was interesting for me was, again, to enable another platform with Arduino so that you could have another-- I guess what's important for me is that the maker movement has multiple possibilities, that you are not locked forever in one product.

But if you need to move, if you want to move, if your code now could work on an Edison much better because the Edison allows the feature that you need or whatever, migrate it to Edison. If you have a device that is running Windows 10 for IoT, and you have a bunch of Arduino code that you already wrote,

and you want to port it to that device, OK, cut and paste. So I wanted to enable people to move from platform to platform, and I think Intel, Microsoft, they are part of this idea that you can open up.

Hodge: At the same time, you're coming out, I believe, with the Zero, which is sort of a more powerful version of the original Arduino. There's not really network connectivity or anything, but if you want to have a standalone board that's very powerful.

Banzi: So what's happening is that, obviously, the market has been driving a lot in the direction on the microcontroller side. People have been pushing a lot for having more ARM-based parts. And one of the parts that I think a lot of people are excited about is this Cortex M0+, which is like a very lightweight core ARM microcontroller which is designed for connected applications because it's very power efficient. There's a number of features in there. But you can have it for very cheap.

And so we started to work with Atmel to actually port the code from the classic Arduino over to this one. And we actually spent quite a bit of time on the project because we wanted to get it as compatible as possible with the old one, which means quite a lot of work because a lot of Arduino libraries are written only thinking about the old Arduino Unos. So they make no effort to stay cross-platform. They just talk directly to registers on the processor.

And this reminds me about something I read many, many, many years ago when I started programming my first Mac, with the MPWC or something, they were saying, do not code these functions directly because otherwise your code will not be compatible. So they were having problems with people accessing some addresses on the memory directly, and then when they changed from a mode to another, the application would break.

So I guess this is what developers have been doing forever. And we try to provide them with nice APIs, so we are doing quite a bit of work to do that. And we believe that there's going to be a lot of-- now a bunch of products are going to be based on Cortex M0 plus a connectivity on a module that you can use to bootstrap a product very quickly.

So you do your programming on the Arduino Zero plus the Wi-Fi shield on top, and then in parallel, there will be a module with Wi-Fi, processor, and maybe some power supply on it. So that if you want to then turn your prototyping 100 units, you buy 100 modules, and you build it. Clearly, there is an issue that a lot of these parts are SMD, so they are not available to be placed in a breadboard, so there's quite a bit of work to do that, to put those parts on them.

Hodge: That was something I wanted to ask you. You had mentioned earlier when you first designed these, you were using things you could find through-hole [leads] so people could solder. Now it seems like there's a lot fewer parts. Companies like SparkFun, a large part of their business is just providing breakout boards for SMDs. Is that making it harder for hobbyists, or are the prepackaged solutions getting them to where they need to go?

Banzi: Well, I think in general, obviously the moving towards SMD parts is making it more difficult for a lot of hobbyists and makers because they don't have the skills to solder SMD parts. But on the other hand, I think where the maker movement and Adafruit and SparkFun and a bunch of other people-- they brought thousands of breakout boards. Especially if you look at a case like Adafruit, they would take a sensor, put it on a breadboard, create a very nice tutorial on their website, and create a software library for Arduino that encapsulates the whole functions of that sensor and put it on the GitHub.

You would buy their product because, obviously, you get an accelerometer. You buy it. There's an example that works out of the box. There's a library. There's a tutorial. Boom, in 15 minutes I'm working. I have my own thing. And you see sometimes, some projects built by people, there are an assembly of a bunch of these little breakout boards connected together to form some kind of a product.

Hodge: I have some of those sitting in our lab [at the speaker's workplace] right now, specifically using Adafruit sensors. How many companies like Adafruit and SparkFun are you strongly partnered with, where they're really part of the community?

Banzi: So we do quite a bit of work with-- so obviously, we have SparkFun manufacture some Arduino-branded boards. Adafruit has already a couple of boards. And also, we tend to work with them. We are doing more and more with the Seeed Studio, which is out of China, based in Shenzhen. They really, really understand the maker movement, but I think they represent an interesting evolution of the Chinese model.

Before, China was known for copying and making it cheap, while Seeed Studio is about innovating and working with people providing value. So it's a very interesting company, and we love them. And also, there are people in Europe we collaborate with. There's distributors in France and Germany we work with that make lots of modules. It's a very big community. Every time I go around the world, I bump into people who are doing things. I love that.

Hodge: We talked about the European and the American maker community, some of the similarities and differences, but what's the maker community like in Asia?

Banzi: So I don't know very much about it because I've only been to China a few times, but I went to this maker carnival in Shanghai, and there was a lot of little startups made by makers in China doing all sorts of things. So they do a lot of interesting things. I think a lot of them are still in a phase where they look at the classic American startup, and they're trying to apply that to what they're doing.

So they're still in that phase where they are basically using the model that feels more exciting as a starting point. But we can expect them to start producing a lot more exciting innovations very quickly. A lot of Chinese smartphone companies now are starting to challenge some larger companies because they're starting to not only copy but innovate. Same is happening in the maker movement.

Hodge: And how much is-- for lack of a better term --cloud computing impacting you? I thought I saw an announcement recently they you now have server cloud-based IDE at least for some products? Is that becoming more and more popular?

Banzi: Yeah, when I started working on Arduino, my objective was that my student wanted to build devices, and back then, connectivity was kind of a side thing. It was too complicated. And now, makers we work with, they want to build a lot of connected products, so for us, it's about providing the tools for makers to be able to build these kind of connected products. So it's about hardware, yes, but it's also about writing software libraries that, again, provide a mental model that represents those kinds of devices, produce the content, the tools, the IDE.

So one of the things that we noticed, that a number of people liked the idea of having their IDE in the cloud. We wanted to create a platform that would basically act as an aggregator of different things that people do on the website in separate areas, bring them out in the same tool. And in fact, the tool is called Create, Create Arduino, as we say. And so, the idea is also that once you write your code, very soon you'll be able to see all your network-connected Arduinos, so you could push the code to them over the network.

So it's about more and more providing these kind of tools that make it very simple. I want an Arduino user to be able to build a meaningful connector device with 12 lines of code. It can be done, but it's about saying, this is the mental model. This is how it works. You cannot build everything. You'll be able to build this kind of stuff very easily. And in Arduino, this works in the past. We said, this is a mental model to develop this kind of stuff. It doesn't apply to everything, but it will work very well for this majority of use cases.

And then, people took that and stretched it to do everything else. But if you try to be too generic as a platform, you basically don't convey any message, and people get lost. If you have a very strong voice, people then stretch maybe what you do to cover something they're trying to do, but the voice is there. The idea is clear.

Hodge: Talking about stretching the platform, just a few technologies I wanted to throw out your way. So you've done a very good job of democratizing the microcontroller, and I'm kind of curious as to if there's some other technologies you're interested in democratizing. So one thing with Arduinos, particularly if you're using Wi-Fi for connectivity, your battery life can be a big problem. So have you been looking at mesh network, software-defined networks, ways of creating lower power devices? Is that something you think will catch on?

Banzi: Yeah, so for us, to begin with, we know that now is the time where we need to sit down and write some Arduino APIs that make it simple for people to understand power management. Because a lot of products at the moment, Arduino products, don't have product management-- power management, sorry.

If you connect the battery, it runs out in two hours. So we have to actually wrap that into a library that teaches people how to do power management. For example, a lot of the microcontrollers we use from Atmel, they're considered the best in their class in terms of power consumption, so it's really about the software and the design of the port to bring that out. Obviously, there is a number of technologies.

Now, we're trying to make it very cheap and simple to do Wi-Fi. We're trying to make it very simple to do BLE and Bluetooth 4 and allow people to build connected devices that connects to a phone. The hardware is there. It's a question of how do you wrap around that a conceptual model, APIs, ideas. Then, there are other areas where I'm interested. I'd like to really pursue this-- what can you do when you take very simple Linux machines, like super, super simple machines, very small, very tiny.

And also, another area where I'm interested in-- our software-defined radio, we never really went into that. I think there's some interesting stuff going on, but at the moment, the stuff I see people do is more about using those USB TV receivers, plug it into a computer, and then using it to listen to airplanes or whatever.

Hodge: Or satellites.

Banzi: Or satellites, yes. So one area that I'm very, very interested in is-- apart from everything cloud and everything else, OK --I'm very, very interested in FPGAs. I think it's time that somebody democratizes FPGAs and makes an Arduino of FPGAs.

Hodge: That's excellent because that was my next question. Can you make an FPGA easy to use?

Banzi: So the funny thing is that, the first time I met Tom Igoe, again, co-founder, was in a Starbucks in Union Square in San Francisco in 2004. And the first thing we said, somebody really needs to make a tool that makes it very easy to use FPGAs. Then, in the end, we made Arduino because that was too complicated. But I think now, the tide is turning. There are cheap FPGAs that are very powerful. The companies that make FPGAs are more and more open to tools that are open source. Or now cloud computing allows you to have the compiler online and just use it as a service.

We have actually been talking to some people about, how do you create the tool that really lets you build a piece of hardware simply? And then, you can almost synthesize your own processor and in your own peripherals and everything else. And how would you create the tool that would allow you to press a button and get a finished product out of that? Can you automate the process of going from your prototype to a consumer electronic device? So I think that's a big, big, big area of building tools that allow you to go from a prototype to receiving a PCB that's finished and working, but you don't know anything about how to design a PCB or anything else. So FPGAs could have a part in that.

Hodge: OK. One of the things I've always noticed that's very powerful about the Arduino as an educational tool is that it's so basic. It's sort of like how people used to be able to have like an Apple II, where you had to be concerned about the low level impact of things. You didn't really have-- I'm showing my ignorance of Apple II's, but you don't, on an Arduino, have an operating system, so you have to deal with interrupts directly. It teaches you a lot more about the computing hardware than you get on a modern PC.

Banzi: It is actually used in a number over university classes to teach about computer architecture because it is a super simple RISC processor with very few instructions. It's very neatly designed. Either the AVR or the ARM, they're good architectures to study. You can buy them as a cheap Arduino product. Since the software is all open source down to the last line, you can actually peel all the layers and actually go through and get to the point where you actually understand how everything you do impacts the low level.

So I think a lot of people like that, the simplicity of that. And they use it to learn about AVR or ARM architecture, and now, also Intel architecture has become more accessible because you have this kind of super direct line down to the hardware. So it's being used a lot as a teaching tool as well, yeah. A lot of the people who use Arduino, they might not even use interrupts. They might not even know that they exist. They might stay very high level, and it's enough for them.

Hodge: So you've recently had celebrated ten years of Arduino. What would you like the next 10 years to bring?

Banzi: Oh, wow. It's interesting. It's not easy. I don't know. I would really like to figure out ways where I can enable more people to do more with technology. So one of the things that I mentioned, I would really like to work on a project that would allow me to build tools that people can use to build something that works with very little knowledge about the inner working.

And if they say, I want to have now 100 of this, there's a way that they can just press a button, put a credit card because hardware, somebody has to pay for hardware. Because it's physical. It's atoms. And at some point, they get a box with a working circuit that represents what they did in their prototype as a thing. So enable people to really participate in the innovation process, so if there is something that the current market says has to be like that, you need to have the chance to say, wait a second, I think we can also do this. We can have a different point of view.

And in a way, I like the fact that a lot of hackers-- in the nice sense, not in the sense that they're some type of people used as nasty people --so hackers, makers, thinkers, a lot of them have this kind of idealistic side, and they are wary of the government snooping on them. They use open source as a way to provide an alternative to what is the mainstream thinking. I like the idea that you can create hardware tools that will do the same, that can provide alternatives to the mainstream thinking.

Hodge: We've gone through 10 years, more than 10 years, of history in a very compressed time frame. Is there anything that I didn't ask but I should've, things I've left out that you'd like to talk about?

Banzi: Oh, boy. Well, you know, ten years is a long time, and it's not easy to summarize in a few hours. I think it's interesting to see the evolution of this. At the beginning, it was a very small group of people, kind of nerds, in a way. And then, slowly, I was amazed to see how it expanded a lot, and the maker movement became big, and we are still at the beginning of the things that the maker movement can do as a way to change the way we think about the devices that are part of our life.

I think the only thing that I see, which is kind of in a way a little bit sad, is the fact that as the market grew, the same thing that happened in other markets happened here, so you get more-- as the market grows, there's more money to be made, and then a lot of sketchy characters move into the movement. So in a way, it is something that was expected, and it's happening now. And I guess it's a consequence of the

success, when something's a success and might lead to money. So I think I would like to keep working to try to keep a piece of that world, a little bit pure and a little bit childish, with the same kind of creativity and enthusiasm that a child has.

Hodge: Do you think there's become too little emphasis on design and maybe a bit too much on technology in some of the applications?

Banzi: I think in general, this is what's happening to the connected device IoT world. There's a bunch of products that come out which come out because the technology is there. But there is not enough thinking on the design and then the product and the impact on how that would impact people's lives. So we have to go through, I guess, a bunch of products that are going to be a little bit, eh. A lot of technology, but not enough thinking about the rest.

And then, I think a number of people will start to understand that type of products and start to come up with products that are much better balanced between the technology and the way that really impacts people's lives. And there's a whole issue of people at the moment, they don't understand that connected devices bring a lot of positive things to your life, but they also can bring a lot of negative things to your life.

The security, safety issues are a lot. My phone tells me, this application wants to check your email. Do you want to allow it or not? But if the device doesn't have an interface, how is it going to tell me-- I'm going to reveal this information to this other person. So there's a lot of issues we have to solve there. There's a lot of possibilities and a lot of bad things that can happen.

Hodge: What would be your advice to young people of high school age thinking of getting into this general area, possibly as a career, physical computing, connected devices, that sort of thing?

Banzi: So my suggestion is that the most amazing people I work with in my life have always been people that had a couple of sides to their soul, in a way. They were maybe good at the technology, but they were also into arts, or into design, or they wrote poems, or they traveled, or they liked French cinema, or whatever. So in a way, I think it's important to love the technology for what it can give you, but it's important not to be completely focused and immersed in technology that you lose contact with the real world.

You need to have, I think, a balance between the technology and the more artistic cultural side. So you need to keep an eye on both. Because there's a lot of people that get so into the technology they start to lose touch with the rest of the world. The people who are into technology are a finite amount of people, and there's other people in the world that don't have the same ideas. In a way, I think it's important to be a citizen of a couple of worlds, not just the technology.

Hodge: And any other concluding thoughts that I've missed?

Banzi: Well, you know, for me it's incredibly exciting the fact that I get interviewed for oral history at the Computer History Museum because I read sometimes some of these. I was on a plane a few weeks ago reading the interview to Federico Faggin about the creation of the 4-bit microprocessor. So in a way, for me, I'm incredibly honored that this is happening.

Hodge: Well, you're welcome. It's our open invitation to everyone. If you change the course of computing history, you get an oral history here. So with that, thank you for your time.

Banzi: Thank you.

END OF INTERVIEW

