Intro

{SLIDE: opening page}

2 subjects:

1. The ArduGuitar Project
2. Everything that went around the project, the meta-project.

I am not going to define all the terms and references as I speak. giyf.com can do that for you.

I am going to speak for 25 minutes and will take questions at the end. Please hold your questions for the end, or see me later.

I will not be playing the guitar on stage here - there's a limit to how much of a fool I am willing to make of myself.

But, I might make a small demo if there's enough time.

In any case, demos will be given later in the hardware hacking area.

Who I am

I am characterized by 2 things:

1. Lots of dreams
2. very very little money.

Having very little money, and necessity being the mother of invention, lead me to learn how to get by without buying lots of stuff.

By this I mean scrounging around and fixing broken things.

I think that today there are very few broken things I could not fix, if I really wanted to. I spent all my life learning that.

Along with learning to fix things, I learned to fear breaking things that I would not be able to afford to replace...

Fear is a bad thing.

The song says, "When you've got nothing, you've go nothing to lose." But it's more like, "when you've got nothing, you can't afford to lose anything!"

So being poor was a mixed blessing:

- it forced me to learn lots of stuff,
- but it made me scared of losing stuff, too.

But there is one other thing that characterizes me:

- Tenacity. I never give up.

This too can be a mixed blessing...

My Vision

So back to the why I am talking to you.

In my poverty and dream to play electric guitar, over time I had found cheap broken guitars and tried to fix them.

I learned what an electric guitar was made of:

{SLIDE: an electric guitar photo of my Godin}

- **pickups** which capture the vibration of the string and output an audio signal (whatever that means),

- **switches** to route the signals from the pickups,

- **Volume controls**: potentiometers,

- **Tone controls**: potentiometers + capacitors,

- **output jack:** where all the signals end up and go off to the amplifier.

So there wasn't much to it.

At this point my understanding of electronics could be completely described by:

{SLIDE: V= IR}

- *V = IR*

That's all I "knew." But even that I hardly understood.

This was the state of my being for around 30 years.

In the meantime, I had gotten a job and had revenue that allowed me to do more than dream.

I finally started taking guitar lessons, and got a serious guitar.

The guitar was too serious for me to wreck so I brought it to some luthiers (guitar makers) for adjustment and became friends with them.

From them, I learned that even top-end, ultra-expensive electric guitars were still made exactly as they were in the 1950s.

In fact, some guitars with 5 figure price tags were put together by people who could hardly solder...

...

I should tell you that for a long time I missed the 1960s.

I'm a hippie since my teenage years....

I thought that the 60s were as good as it could get, and in some ways that remains true today...

But, I love the 21st century. I love what modern technology and Internet have made possible.

I developed a vision of a 21st century electric guitar.

It should maintain the audio qualities of the 20th century guitar, ie. analogue sound from pickups.

But it should be controllable numerically, programmatically, remotely, wirelessly, etc.

{SLIDE image of the guitar with buttons and the the guitar with a tablet next to it. }

I wondered why guitar electronics had not advanced. I discussed this with many pros in the music business: players, builders, engineers,...

The universal reply comprised 2 points:

{SLIDE : Why not done yet?}

- It would be incredibly difficult and expensive to do anything other than what we have today.

- No one would ever want anything other than what we have today.

Depending on the person, one or the other of those reasons would be underlined.

As the greatest mind of the 20th century said, "What do you care what anybody else thinks?" (if you want to know who said that, giyf.com).

So no one wanting my idea didn't matter. I wasn't trying to sell it anyway.

But the incredible difficulty part was another story. I knew nothing about electronics and had no idea where to begin. I also had plenty of other things to do with my time.

For a year, the project laid dormant.

Inspiration

{SLIDE: Inspiration: TED: "the community of inspiration"}

One of the things that makes the 21st century a great time to be alive is the rise of communities. Dispersed communities, living on Internet's long tail. (If don't know what the long tail is? giyf.com)

How many people here have heard of Marshall McLuhan? Show your hands please? Can you name any of his books?

Recently, I was re-reading his seminal work "The Media is the Message" in the Brussels Metro. Even today, it seems so wildly subversive that I kept looking over my should to see if anyone was watching me...

Luckily, only the NSA was there - aren't they doing a Great Job !?

In 1962 Marshall McLuhan predicted the arrival of Internet:

{SLIDE: Marshall McLuhan}

"In this new age, humankind will move from individualism and fragmentation to a collective identity, the global village."

He was a profoundly forward looking thinker... He also wrote something which was directly applicable to my situation:

"When faced with a totally new situation we tend to always attach ourselves to the objects, to the flavor of the most recent past. We look at the present through a rear-view mirror. We march backwards into the future."

That means that whenever we see something that is new to us, we evaluate it relative to our past, and in particular to our recent past experiences. This can lead people to misunderstand, or to avoid innovation.

An example of this comes from what may be just a made-up conversation but it illustrates the point:

Henry Ford was asked to explain how he knew that people would want his automobiles, did he ask them about what they wanted before he made them?

He replied,

"If I had asked people what they wanted, they would have said faster horses."

Ford wasn't interested in horses.

–

All that thinking was very *"a propos"* to my project:

- it was too different from anything to which people could refer, so they couldn't imagine it,
- it wasn't a faster horse.

A long time had gone by since I had the idea to make a 21st century electric guitar. I had stopped actively pursuing the problem and was thinking about other things.

Then, at the end of 2012, I went to the Brussels TEDx and that made all the difference.

Mitch Altman spoke at that event and his talk was so inspiring to me that at a break I talked to him about my idea. In a minute he convinced me that it was both a good idea and very possible. He told me to try the Brussels hackerspace community for help.

Is Mitch Altman in the room? If he is, please stand up, Mitch.

{SLIDE: Thanks Mitch!}

Thank you for making all this possible.

Tech Talk

That was the end of the meta-project talk.

Now, I'm now to talk about the actual implementation of the ArduGuitar and tell you about the what I actually built.

Please excuse me for any errors I may make due to my limited understanding of all this stuff.

So remember the Electric Guitar circuit? A switch, 2 pots and a capacitor.

The real problem was how to make a digitally controlled version of a mechanical analogue potentiometer.

{SLIDE: pot circuit}

Really, that's all that is needed to make a digitally friendly electric guitar. Just replace the old fashioned mechanical potentiometers with something which can add resistance to an analog voltage while being controllable from a digital device.

But, how could that be controlled digitally? I had no idea.

So I asked a question on the Belgian Hackerspace mailing list. At first, it seemed like there was some kind of communication problem.

Ever seen the film, "Cool Hand Luke"? Paul Newman's acting masterpiece?

"What we got here, is a failure to communicate."

LED-LDR

But, after a few iterations, and lots of patience on the part of the hackers, someone, in fact a brilliant scientist named Johannes Taelman, suggested using "LED-LDR" to make a digitally controlled variable resistance.

By the way, Johannes is currently in the hardware hacking hall showing off his own really cool project and I strongly recommend visiting him there!

I had no idea what that a LED-LDR was, but I learned that it was actually 2 things:

 • a LED to shine light,

 • a Light Dependent Resistor that changes its resistance depending on how much light it receives.

Although it took me several weeks to understand enough about what that was and to see how to use it, I soon realized that 2 of these LED-LDRs could be used to make a potentiometer!

{SLIDE: Pot + LED-LDR}

I had gone through Banzi's book *"Getting Started with Arduino"* so I sort of knew how to control the brightness of a LED …

I was in business.

And I could even use a LED-LDR as an on-off switch, so all the problems were solved!

At this point, someone on the hackerspace mailing list told me,

"It will certainly work to some extent, and even if the result is too noisy or bad quality to be used in music, it will still be huge fun to make it."

I mention that, because it was a bit worrying. I hadn't considered "noise" at all. My simplistic V = IR understanding didn't contain any notion of noise.

I was going to pay for this later on...

By the way, LED-LDRs come as pre-assembled components. There are several types with slightly differing specs. They are called Vactrols and the best ones come from Perkin-Elmer.

Secondary Issues

Once the key problem of making a digitally controllable analogue potentiometer was "solved", sort-of, the rest of the issues in the project needed to be addressed. In particular the overall design.

I read the great book *"Making Things Talk"* by Tom Igoe and learned about serial communication and Bluetooth. It seemed straightforward.

My conceptual idea was established:

- the guitar would have an arduino embedded in it, with the Vactrols, and a Bluetooth module,
- a bluetooth device, such as an Android phone or tablet, or even a computer would have a GUI to send messages to the Arduino and control the guitar sound.

It seemed simple enough.

And it was!

{SLIDE: schematic}

This is the schematic of what's in the guitar I have here.

- Upper left: pickups, note single and double coil
- Center top: 4 switching Vactrols, LED is just full on or full off, no PWM
- Center: orange raw signal bus, heading to the volume and tone control vactrols
- Center lower: Volume and Tone control Vactrols, Note that a 3rd Vactrol is used in the volume control to shunt and get maximum volume.
- Left lower: blue output bus, with tone capacitor to ground part of the the tone low-pass filter,
- Lower left: output jack to cable to amp; Note the amp ground is the guitar ground; the Arduino & 9V ground is isolated by the opto-isolating Vactrols.
- Right: the Arduino and Bluetooth module and 9V battery.
- lower: the current limiting resistors for the Vactrol LEDs. Note: the R values for the Tone & Volume controls were determined empirically, and seem to be incoherent with the vactrol specs from the Data-sheets... I have no explanation.

Being a pretty good programmer, I got this coded & running quite quickly on a breadboard.

I got an old Ibanez RG-140 and took out all the circuitry.

I connected a very cheap JY-MCU bluetooth module to an Arduino micro which would fit in the guitar's electronics cavity with the vactrols.

I wrote a GUI in Pyside (Python+Qt) for my Linux laptop and everything was cool.

This had taken 2 months. I felt ecstatic!

When I plugged in the Guitar for the first time and it worked... I felt like Dr. Frankenstein when the lightning brought life to his creation!

It was a great moment!

Now, all I had to do was port the python gui to Android!

Oops...

This is when I began a roller-coast ride, yo-yoing between great hope and total despair for a real really long time.

It turned out to go from Python on linux, to Java on Android was a big step!

I do not know Android, nor the java framework for Android.

I tried to make the Python work on Android, but this was very slow going...

Finally, I discovered that Processing could compile to Android-java!

Cool, I thought, progress at last.

But the GUI components and the Processing Bluetooth library (Ketai) were not easy to get to work - and there were some defects.

I also discovered some errors in Tom Igoe's book - when I wrote to him, through the publisher, he admitted to not testing all his code!!!

But in the end, I had a working Processing GUI running on Android!

{SLIDE: technology}

More Problems

Suddenly I had a really cool thing and the same professionals who had told me to forget it a year earlier, now loved it and had "suggestions".

The most obvious suggestion was to *"Get rid of that whistling noise!"*

Indeed, at each bluetooth exchange, a high frequency clicking or whistling sound came through the amp. I couldn't understand how that was possible since the audio and the control circuits were separated by light!

Another good suggestion was to make it possible to automate the changes of configuration. An example would be to make a tremolo effect by going for high to low volume every ½ second.

Also, in showing it to people and talking with them, I found terminology mismatches: for example, I had called the saved settings "configurations", but musicians all asked if that was like "presets".

I changed the name to presets. And made ones that people could relate to like, Eric Clapton's famous "Woman" tone.

The automation was easily done.

Getting rid of the noise was a nightmare.

I asked everyone everywhere for help.

All the forums. I had forum threads with thousands of views...

The Arduino forums were especially friendly, but at the end, no one anywhere seemed to understand what the problem was or how to fix it.

Finally...

Then I thought to ask Bluetooth experts. After all, maybe the 5€ JY-MCU was just adding noise?

So I contacted Sparkfun support to ask about their Bluetooth modules (costing 30€). They were really nice and tried to help but told me that they used Roving Networks BT chips and advised to contact Roving support.

This was the breakthrough: RN support was great, even though I wasn't a customer! They looked at the spec for the JY-MCU and explained to me what was wrong with it and why their chips cost more: Shielding, respecting standards, more capabilities and DOCUMENTATION!

Getting a new Sparkfun Bluesmirf based on the RN42 bluetooth module got rid of almost all the noise!

Suddenly I had something GREAT!

But still not perfect.

I contacted shielding companies, got samples, learned about how to ground shielding...

A Small Miracle

Remember I had implemented automation? Well that turned out to be fundamental in searching for the noise source. By running the coms automatically, I discovered that the gold tip of the guitar cable could act like a stethoscope. The other end was plugged into the amp turned up to full volume. I could sniff around the circuit and see where the noise was strongest.

It was not from the Arduino.

It was not from the RN42.

It was from the Serial TX and RX wires connecting the RN42 to the Arduino!

Those square waves were pulsing out EMI !

I heavily shielded the wires from the BT module to the circuit board and life was beautiful!

And I was invited to a Maker Faire...

And there was a write up on HackADay...

And I was invited to speak here today.

Now let's look at it work.

I'm sorry for the latency between the tablet and the computer, the refresh rate is only about 3 frames/sec....

{Screencast & Demo}

Learn this stuff

Showing your work is really important. Getting feedback, both positive and negative are really helpful.

New ideas come and old problems disappear.

So if you are making stuff, even little stuff, be sure to show it around - and not just to experts.

Show it to anyone who'll take the time to look!

But now that I had something...

I had something to lose...

I found that I had become afraid to make changes in the code, and in the circuit since I had managed to reach a delicate equilibrium state:

- It worked, but I didn't really know exactly how or why.

- So, if I tweaked something, I might break everything....

I did a lot of code review to improve things.

I am still searching for an electronics person to study the circuit and help me understand the noise and shielding issues...

The Future: ArduStomp

Big pro-musicians have sound-board operators and stage help etc. These guys manage the effects, and could use the Android version of the ArduGuitar interface to support their guitar players, but what about people like me - remember it was for me!

A foot-pedal would be so much easier than a touch-screen.

Voilà : the ArduStomp!

{*sketchup* `Enclosure 01.skp`}

With some problems of its own!

I'll be in the **Hardware Hacking Hall** this afternoon if anyone wants to see the guitar and Android interface.

Thank you all for your patience

{SLIDE: thanks}

I'll answer any questions now, or you can see me in the hardware hacking area this afternoon.

To Show the App set this up before hand!

$ adb devices
List of devices attached
015d8bed2b17f410   device

$ javaws androidscreencast.jnlp

right click to set landscape, then resize window

Sorry for the poor refresh rate.

or (but no better)

$ java -jar droidAtScreen-1.0.2.jar

sketchup `Enclosure 01.skp`