

ALESSANDRO CONTINI

selected projects
2010 / 2018

info@alessandrocontini.it

BOSCH RETROFIT KIT

BOSCH RETROFIT KIT - IoT product

The first project I was assigned to while at **Relayr** was the Retrofit Kit. A simple yet powerful concept developed for BOSCH, the manufacturer of home and kitchen appliances.

The Retrofit Kit is an **IoT platform that enables users to augment the capabilities of their home appliances**, when they are not smart yet.

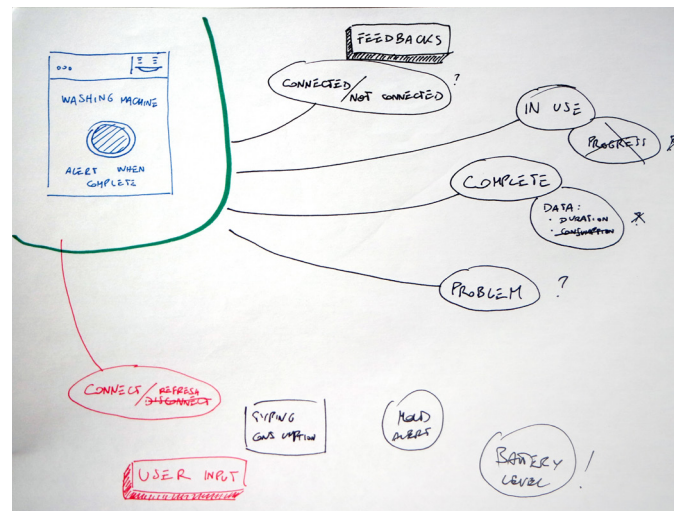
By attaching the hardware to a home appliance such as a washing machine, dishwasher, gas stove, oven, fridge, the users can get data on their smartphone or tablet about:

- power consumption
- program start and end time
- malfunctions
- danger

I lead the design of the physical-to-digital product experience, working side by side with hardware engineers and developers, and **lead the design and prototyping of the UX / UI of the smartphone and tablet app**.

> Sketch, InVision, Zeplin, POP, Gliffy diagrams, Paper prototyping





RETROFIT KIT - IoT experience

UX

Crafting a good experience for a mixed physical and digital product was not easy.

Based on the data gathered from the **User Research set in Singapore**, the first phase of the UX prototyping showed some big challenges:

- **hardware constraints** demanded to come up with solutions to deduct and show users data that was not coming directly from the sensors
- in order to experience the platform at its best, **users had to fill out a lot of information** about their appliances and life routine
- an important part of the digital experience was not happening in-app, but was based on **alerts and push notifications**

Requirements

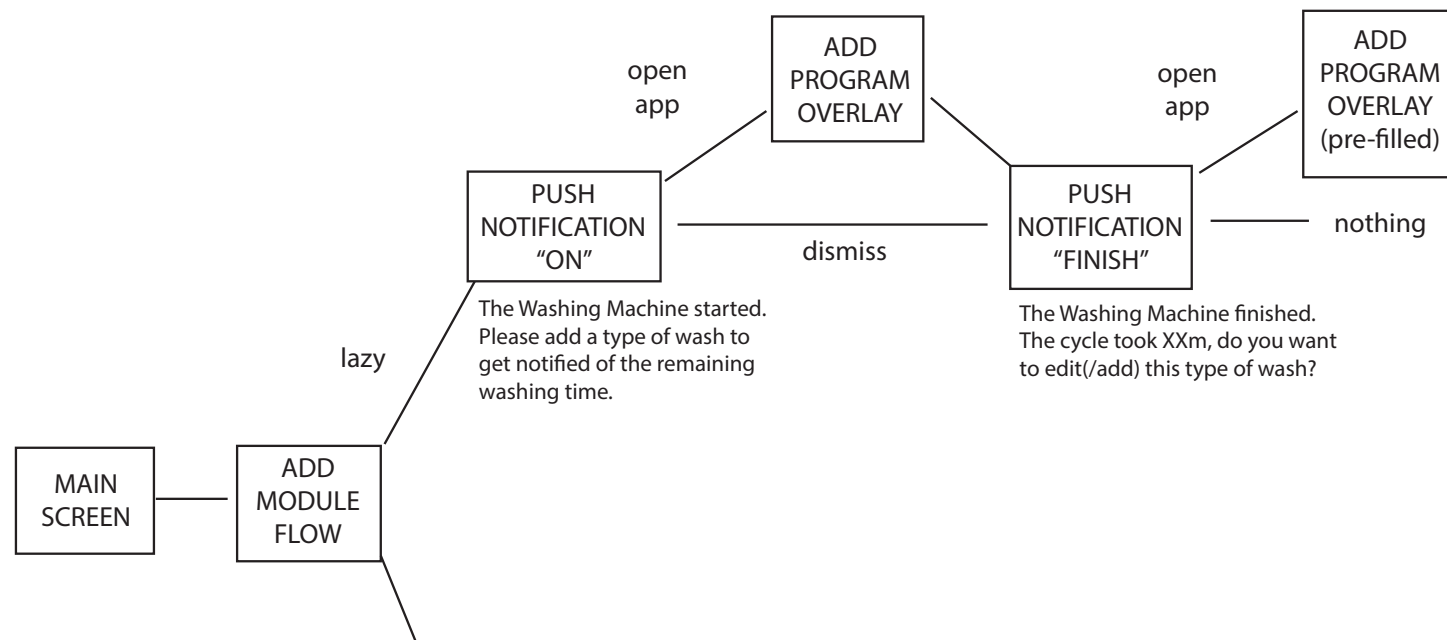
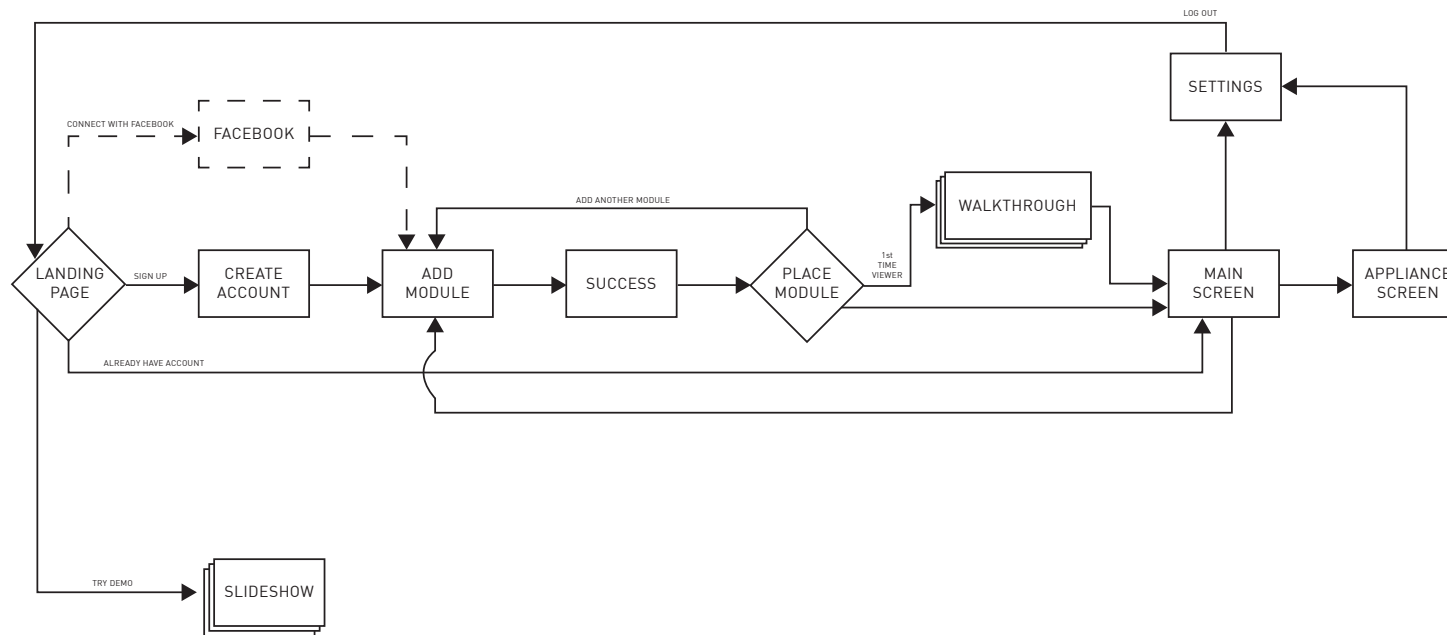
#	Server Team?	Title	User Story	Importance	Notes
1	SERVER TEAM	App Tells User the Washing Machine Cycle Started	As a consumer, I want to be informed that my washing machine cycle has started, So that, I know when to roughly expect the wash time to be completed e.g. 1 hour 20 minutes.	MUST HAVE	
2	SERVER TEAM	App Tells User the Washing Machine Cycle is Finished	As a consumer, I want to be informed that my washing machine cycle is completed, So that I know to take the laundry out to dry it.	MUST HAVE	
3		App Tells User the Washing Machine Cycle is "On-Going"	As a consumer, I want to have status updates, So that I know the progress of my washing machine.	?	• Progress will be displayed locally on iOS Application. All timing calculations will also be stored locally.
4		User Asks the current state of the washing machine cycle	As a consumer, I want to ask for the current status of the washing machine cycle So that I can be sure I have the latest status information (Started, In process, Done)	?	• Application is opened and data is updated automatically.
5		User can select from "default" washing cycles e.g. eco, normal, heavy wash	As a consumer, I want to associate my washing machine cycle in the app So that it knows when to inform me the cycle is completed	MUST HAVE	• Need the content for the default cycles and the fields that are required for the user to input.
6		App can "learn" the wash-cycles	As a consumer, I want the app to be "smart" and learn my washing machine cycles, so that I do not have to spend a lot of time configuring things.	NICE TO HAVE	• This will be handled locally.

7		App generates Statistics of Usage	As a consumer, I want to know my usage statistics about most frequently used programs, wash times, days of the week things are used, months of most frequent usage, So that I can optimise how I do laundry COUPLET STORY As a Product Owner, I want an aggregate of statistics about user usage So that I can make the best products possible for my customers.	MUST HAVE	• Ideally this comes from the Test- API if it is available. • Otherwise, this data will be faked locally.
8		App lets user know a cycle will complete soon as the user leaves the house	As a consumer, I want to be reminded that a washing machine cycle will end soon, So that I can better make a decision to choose to wait for the cycle to complete or leave my house.	NICE TO HAVE	• What is possible with Google API? Can we know the user's current location? What if the user uses a laundry mat and not washing their clothes at home?

← First off we analyzed what kind of data was possible to sense and **display** about **the status of the appliance**. At the same time we defined what kind of **input the user could give to interact with the appliance** or customize its own experience of the feedbacks coming from the appliance.

This helped understanding and creating **Use Case Scenarios** and **User stories** for all of the home appliances supported by the platform.

B/S/H RetrofitKit App – User Flow



RETROFIT KIT - User flows

UX

← I decided to keep the in-app User Flow very minimal and clean, and explode some of the most complex parts.

← One section that required special attention was the managing and input process of the appliances' programs.

We had **two types of users**:

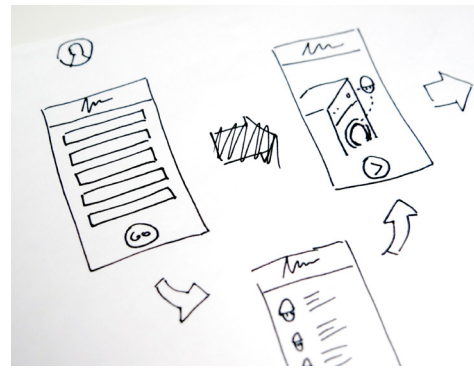
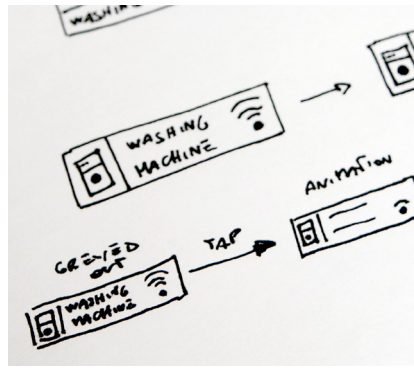
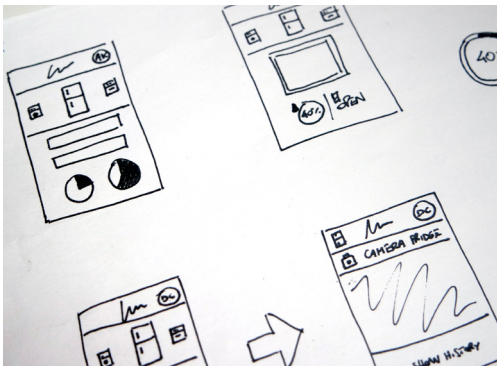
Committed - users that would take some time to input the program details (cycle duration, phases...)

Lazy - users that would initially skip data input requests

For the different users types I designed a different behavior of the push notifications.

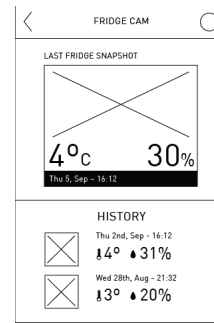
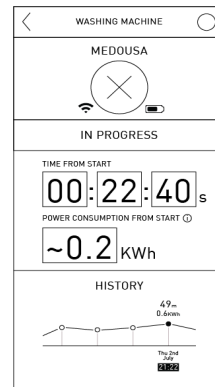
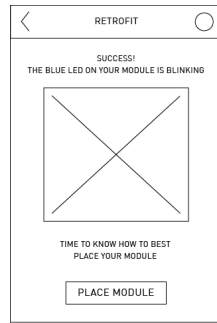
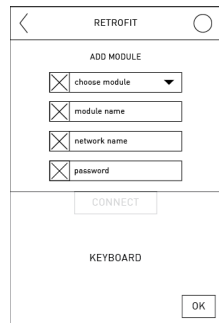
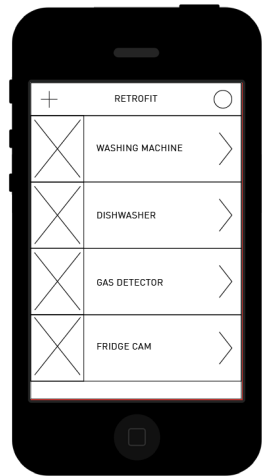
Specifically, the lazy user will be notified when the hardware would sense the appliance starting and will ask to quickly input start data. The same would happen at the end of the cycle, asking to confirm pre-filled data and eventually modify it.

By applying **machine learning techniques** the platform would learn from users habits.



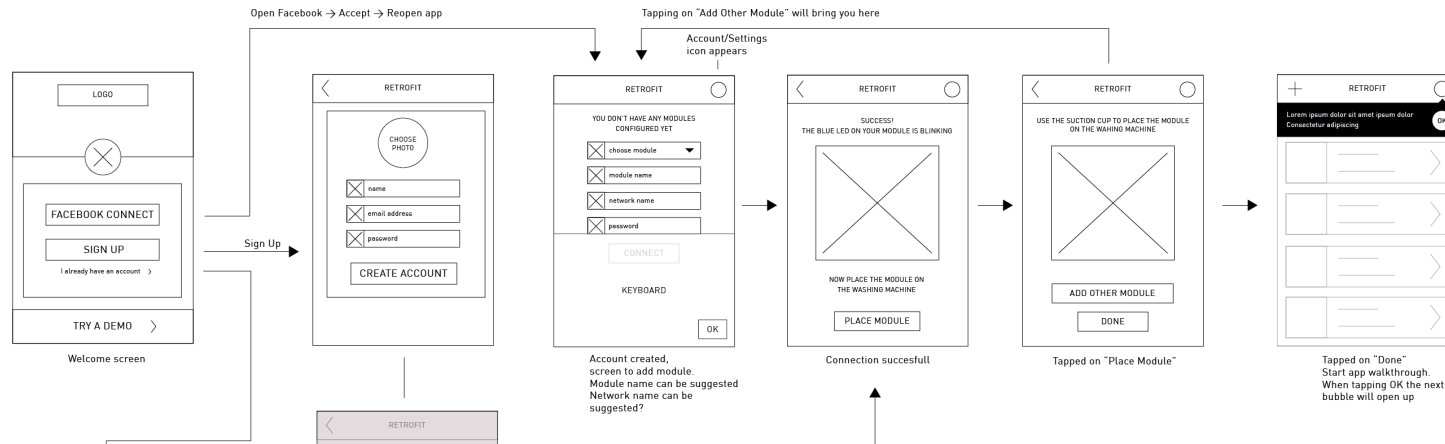
RETROFIT KIT - Wireframes

A very early iteration of the wireframes was sketched during initial meetings. **I always really like to quickly turn ideas and suggestions into sketches on paper.** This way I would collect a lot of material to play with later on.



← After translating the sketches into digital wireframes I quickly built a rough **click-through prototype** with POP App, and ran a round of tests with users that matched the demographics to validate navigation and concept.

Onboarding



← After testing, the Wireframes were iterated and merged with flow details. With this kind of deliverables the developers were ready to start structuring code, layout and navigation between views.

Main View UI tests



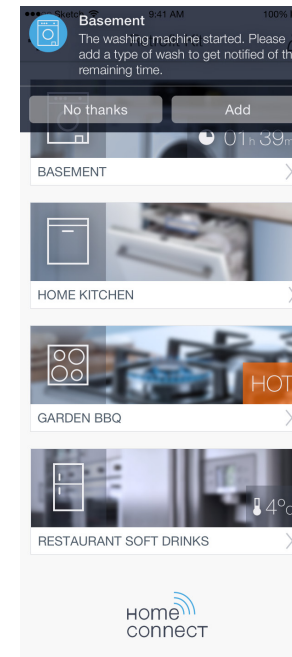
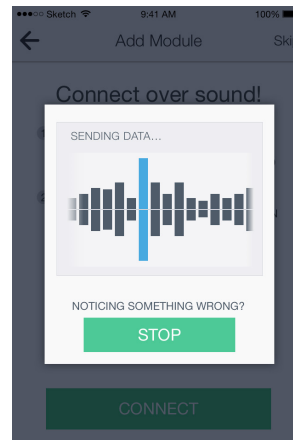
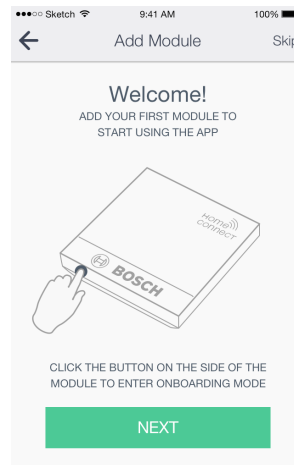
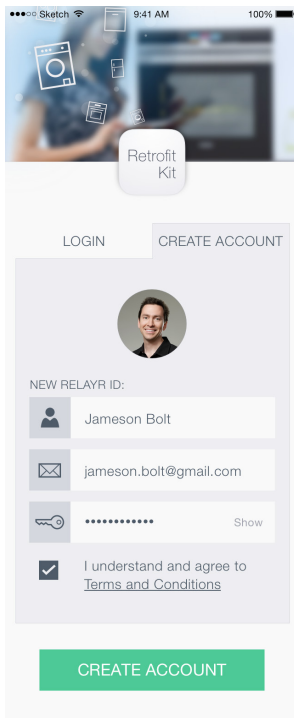
RETROFIT KIT - User Interface



The prototyping of the UI was based on results coming from the first tests and on client's feedback.

← Visually speaking, the UI went through various iterations and was based on a mix of the existing app guidelines and iconography, and a moodboard created after the apps preferred by the users during the user research.

Selection of def Screens

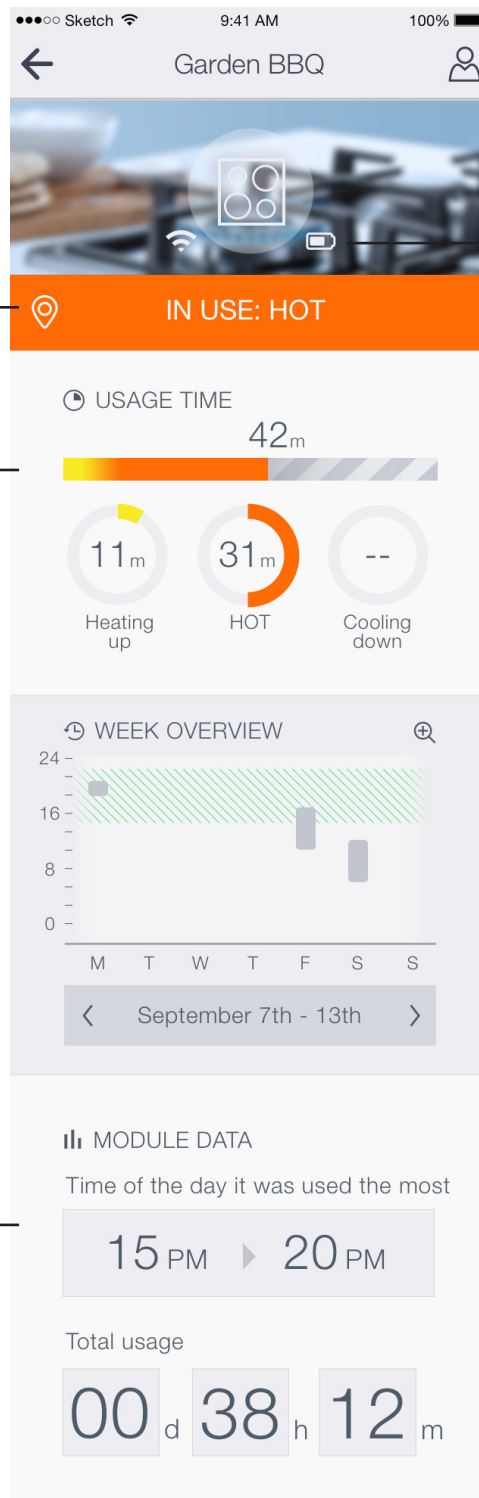


← Main appliance screen walkthrough

Geofencing functionality will automatically enable when user is nearby to notify when leaving a potentially dangerous stove in use

Detailed stats on current usage

Generic stats on overall appliance usage



Status data of the sensing hardware:

- Wireless
- Battery

Color coded feedback for the appliance's status

Browsable weekly overview

ARDUINO CREATE

ARDUINO CREATE – Web platform

While working at **TODO Design** I focused entirely on projects for **Arduino**, leader of **open-source hardware solutions**.

Arduino CREATE is **an integrated online platform designed to enable Makers to write code**, access educational content, configure microcontroller boards and share DIY electronics projects.

The software was initially developed for a stand-alone microcontroller, the Arduino TRE, then extended into the new Arduino CREATE, a web-based platform featuring a **re-designed IDE and a constellation of apps and services to guide the Arduino user from the first steps to complex IoT projects**.

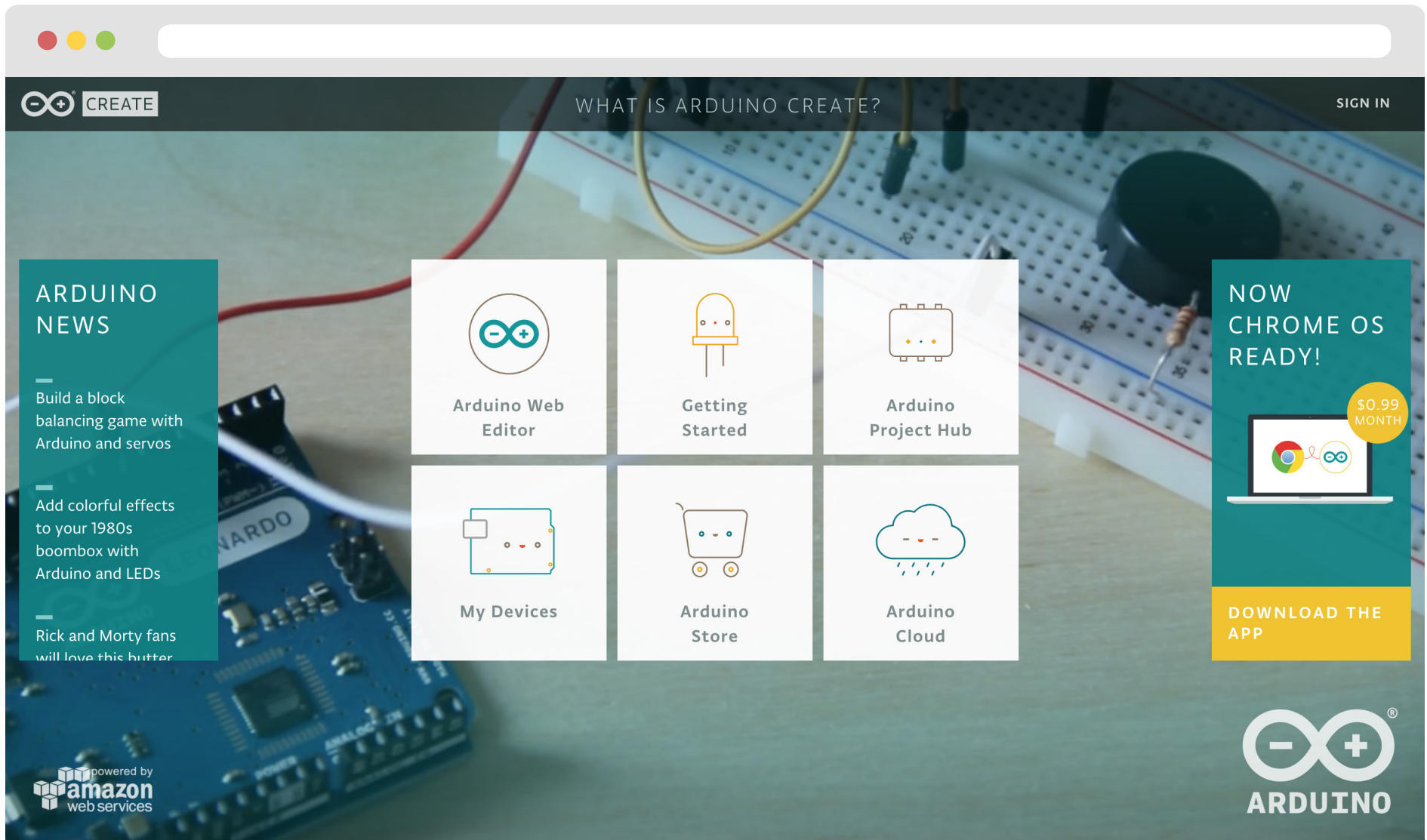
It's been designed as a modular system and developed with Open Source technologies, so that it will allow third parties to create apps for the platform itself.

My role in the project included co-leading the UX/UI design of the platform, and taking care of the UI Frontend development for any of the different sections of the software.

> Adobe Illustrator, AngularJS, HTML5, CSS3, jQuery

<https://create.arduino.cc/>





ARDUINO CREATE – Web Editor

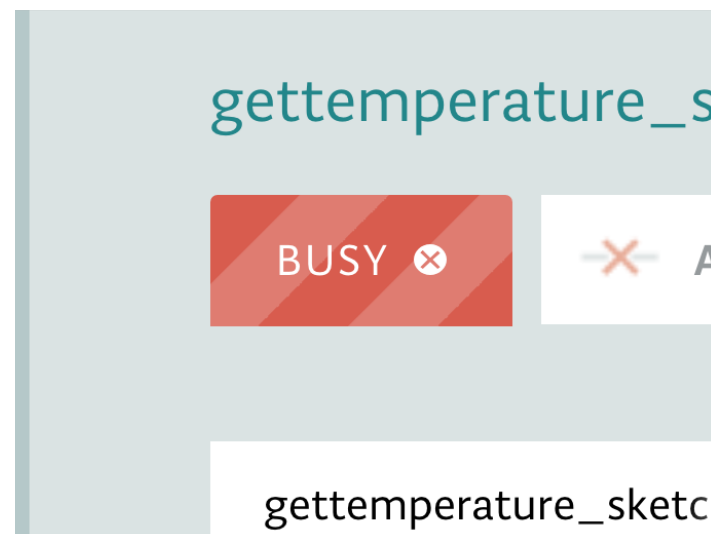
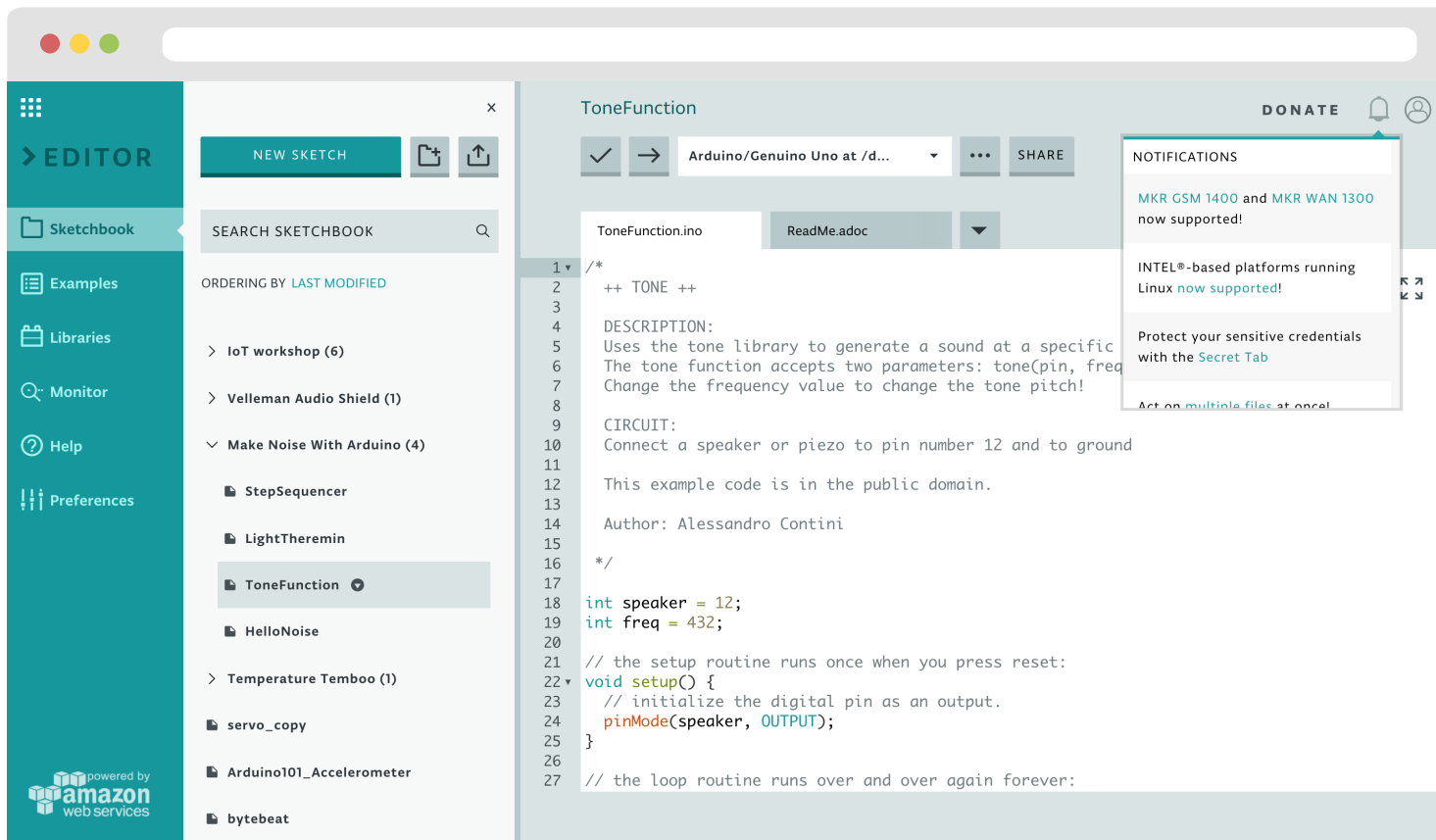
← The Web Editor is the heart of Arduino CREATE. It is **the place where users can program Arduino boards from their browser**.

When compared to the legacy Arduino IDE, the webapp is **filled with new cloud features while still keeping familiarity to the previous software visual language and information architecture**.

To keep this consistency we designed it to feature the familiar code editing functionalities always visible and at the center of the stage. The side menu would instead trigger all the new features in a drawer-like show and hide fashion, letting the user find them when needed.

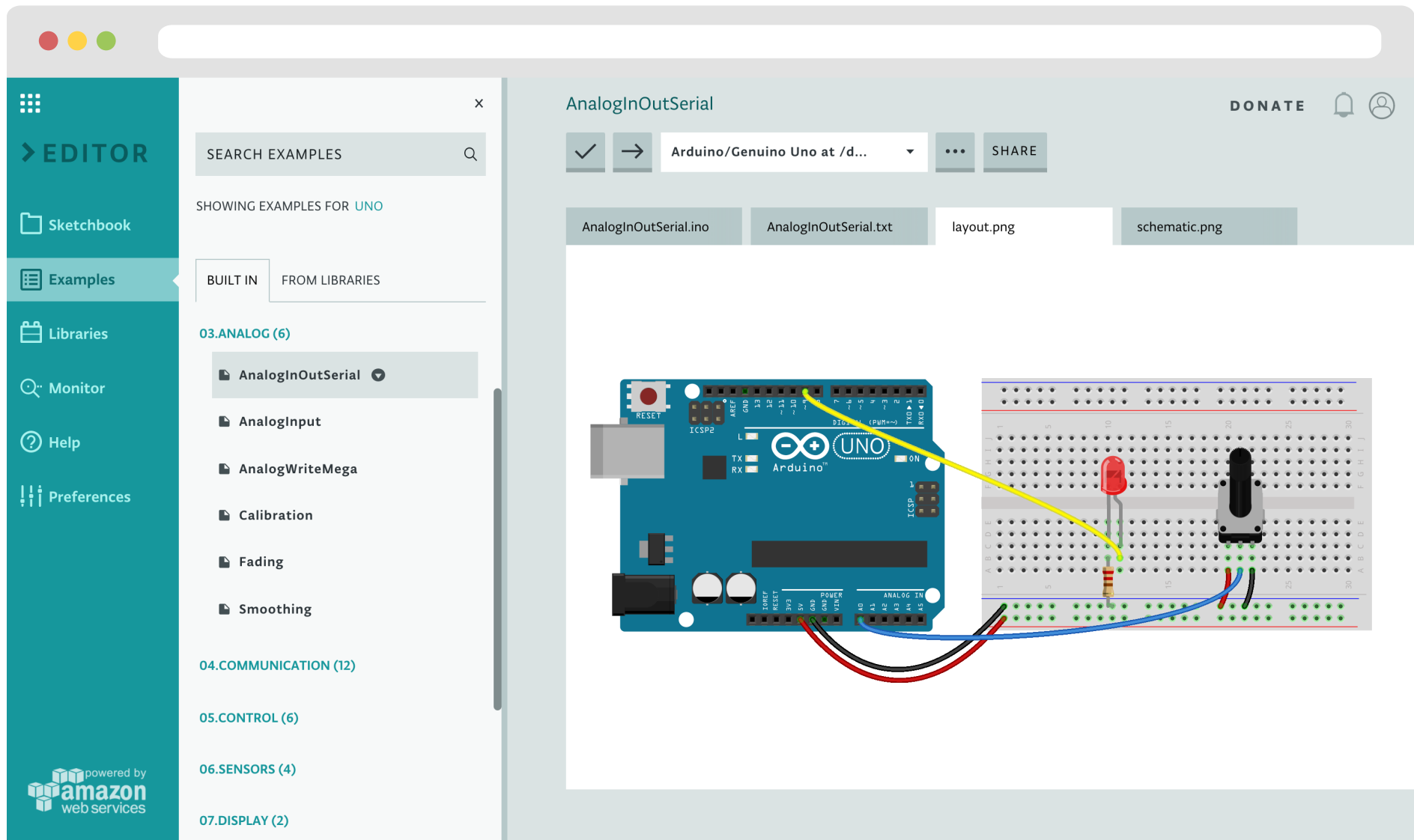
New features allowed users to:

- create, store and manage project files in the cloud
- include and manage featured and external code libraries with a click
- browse tutorials and guides
- upload circuit diagrams and layouts



← Aside from co-designing most of the features, **my responsibility in the team was to translate static mockups into production ready and pixel perfect code**.

I also took care of handling interaction states of the UI elements and designed and coded their behavior and motion.



⋮

> EDITOR

Sketchbook


Examples

Libraries

Monitor



Help

Preferences

 powered by
amazon
web services

AnalogInOutSerial

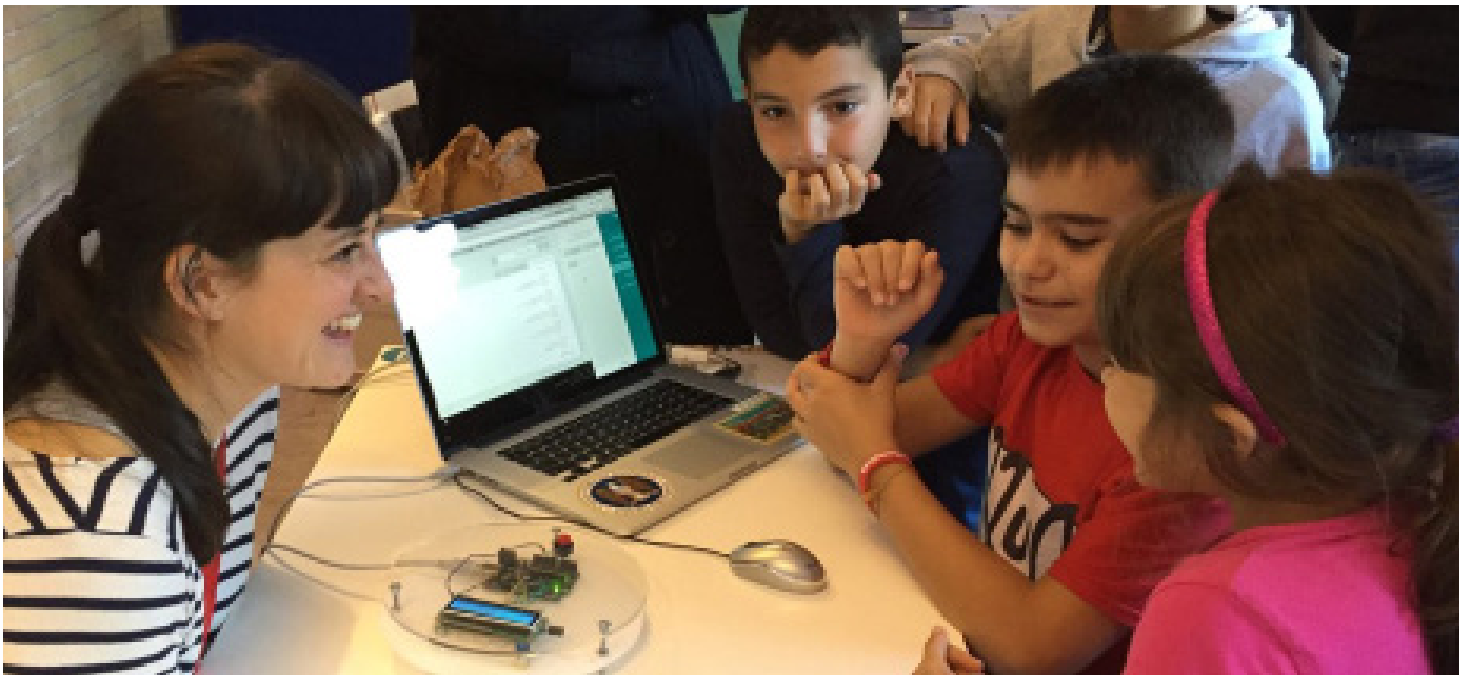
DONATE



✓ → Arduino/Genuino Uno at /d... ⋮ SHARE

AnalogInOutSerial.ino AnalogInOutSerial.txt layout.png schematic.png

```
23 // These constants won't change. They're used to give names to the pins used:
24 const int analogInPin = A0; // Analog input pin that the potentiometer is attached to
25 const int analogOutPin = 9; // Analog output pin that the LED is attached to
26
27 int sensorValue = 0; // value read from the pot
28 int outputValue = 0; // value output to the PWM (analog out)
29
30 void setup() {
31   // initialize serial communications at 9600 bps:
32   Serial.begin(9600);
33 }
34
35 void loop() {
36   // read the analog in value:
37   sensorValue = analogRead(analogInPin);
38   // map it to the range of the analog out:
39   outputValue = map(sensorValue, 0, 1023, 0, 255);
40   // change the analog out value:
41   analogWrite(analogOutPin, outputValue);
42
43   // print the results to the Serial Monitor:
44   Serial.print("sensor = ");
45   Serial.print(sensorValue);
46   Serial.print("\t output = ");
47   Serial.println(outputValue);
48
49   // wait 2 milliseconds before the next loop for the analog-to-digital
```

Arduino CREATE - User research

Thanks to a very committed community, it was not difficult to find users willing to help and collaborate.

Co-creation sessions were organized at the local Fab-lab in Turin to explore and understand the expectations for an online platform for makers.

Paper prototypes were initially created for quickly testing possible features and understand what kind of tools were expected or useful for the platform to really be maker-ready.

Early on, as soon as the bare minimum features of the platform were ready **a beta testing program was launched involving 150 users remotely**. Some of them were cherry picked within the Arduino inner community, others volunteered. We used Basecamp as our communication tool.

Betatesters provided both UX and technical feedback on the platform.



← Testing sessions were also set up at public Arduino events and official Maker Faires. In this scenarios it was possible to stress the platform with more unusual users.

ARDUINO REFERENCE

ARDUINO MANUZIO – Publishing platform

The Arduino Manuzio project is a **Github based distributed publishing platform for the Arduino community.**

Manuzio takes a collection of AsciiDoc files (hosted on a Github repository) and crunches them to create an output in the form of HTML pages and PDF books. As a first step in designing the platform we converted an AsciiDoc file in a book of recipes, an Arduino Starter Kit book.

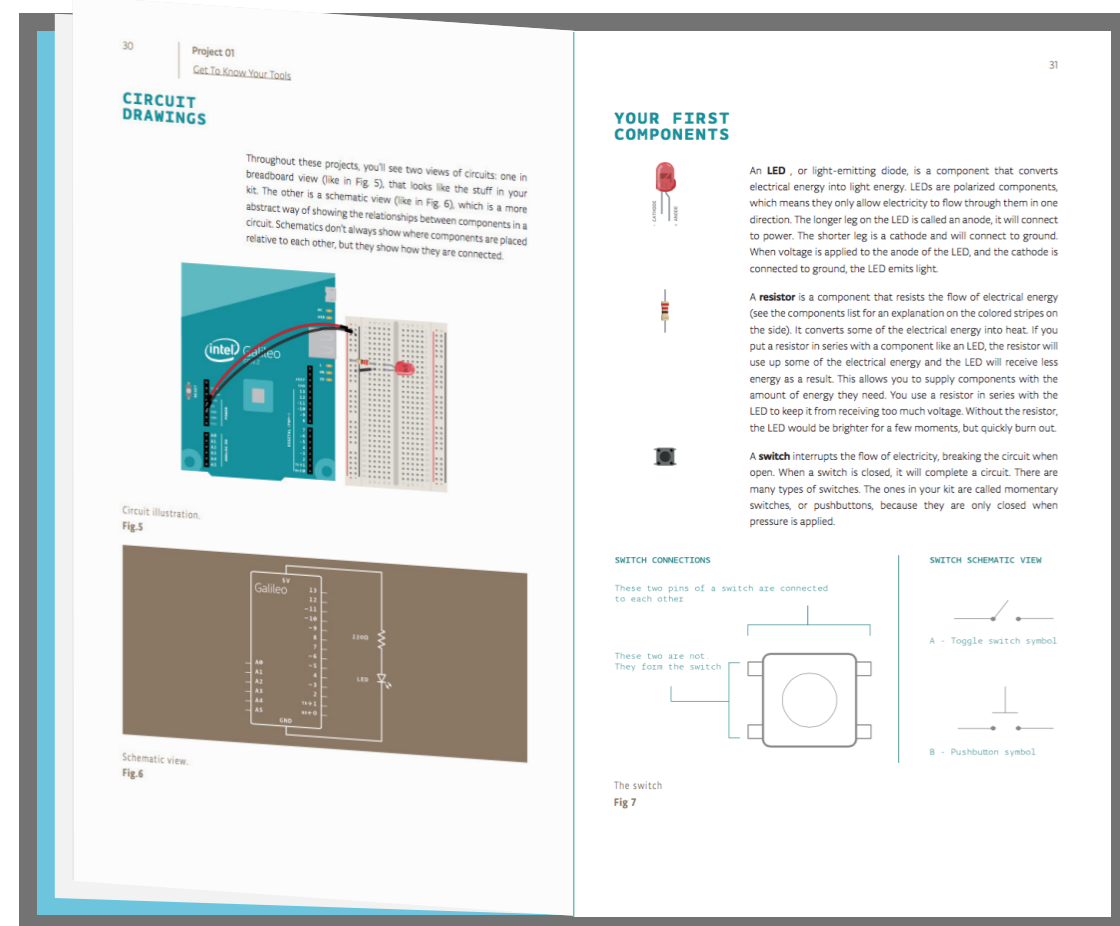
I took care of the rendering of the AsciiDoc code into pixel perfect HTML and PDF files, making sure they comply with the Visual and Typographic needs of the original paper version of the Starter Kit book.

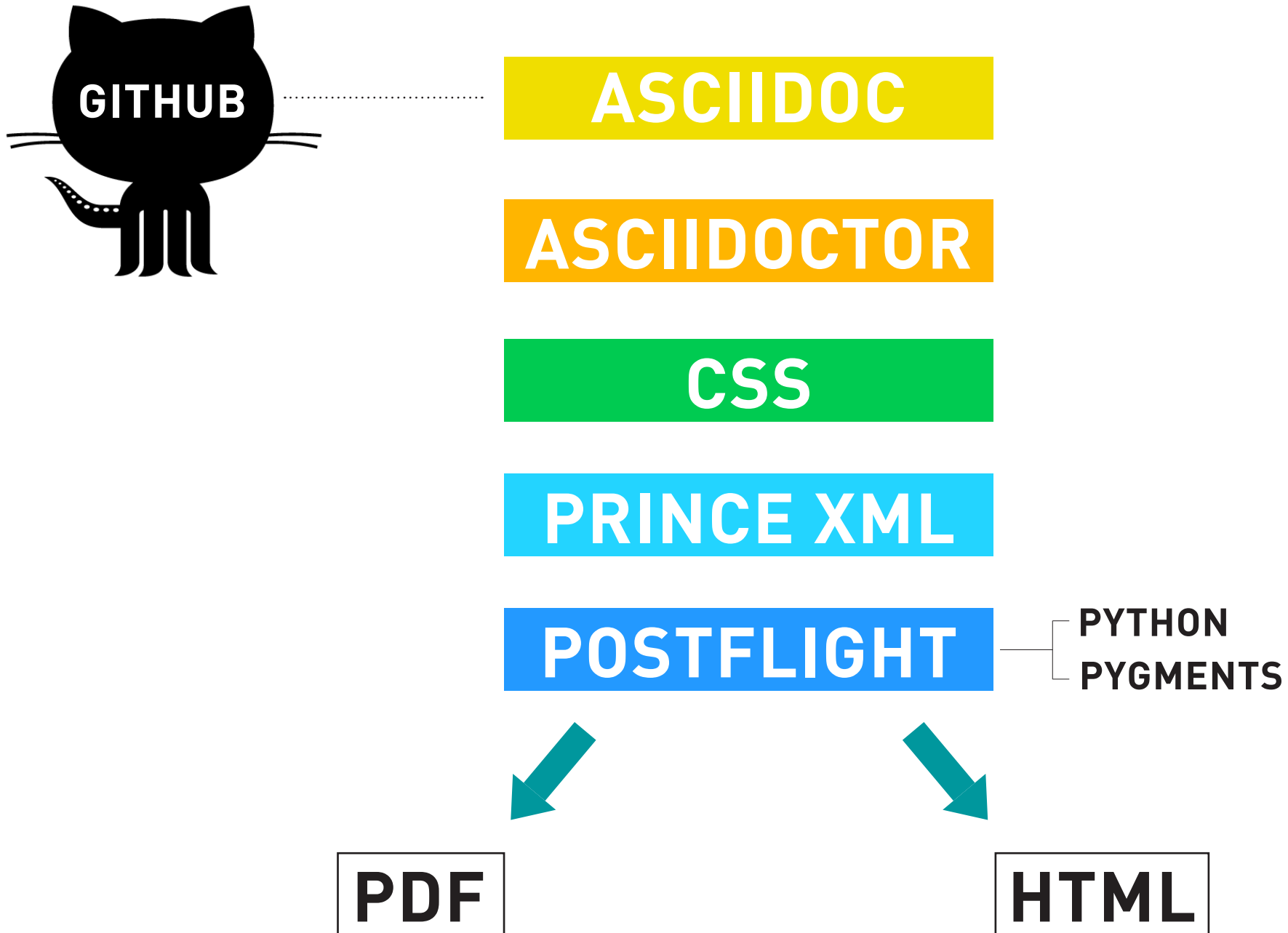
```
chapter_01_get_to_know_... x
105 [role=circuit]
106 === CIRCUIT DRAWINGS
107
108 Throughout these projects, you'll see two views of circuits: one in breadbo
109
110 [role=circuit_img]
111 image::chapter_01_circuit_01.svg[circuit]
112 [role=caption]
113 Circuit illustration. +
114 *fig.5*
115
116 // here a schematics
117 [role=schematics_img]
118 image::chapter_01_schematics_01.svg[schematics]
119 [role=caption]
120 Schematic view. +
121 *fig.6*
122
123 [role=components]
124 === YOUR FIRST COMPONENTS
125
126 [role=side_img]
127 image::chapter_01_side_02.svg[side_img]
128 An *LED*, or light-emitting diode, is a component that converts electrical
129
130 [role=side_img]
131 image::chapter_01_side_03.svg[side_img]
132 A *resistor* is a component that resists the flow of electrical energy (see
133
134 [role=side_img]
135 image::chapter_01_side_04.svg[side_img]
136 A *switch* interrupts the flow of electricity, breaking the circuit when op
137
138
139 [role=img_full]
140 image::chapter_01_full_02.svg[img_full]
141 [role=caption]
142 The switch +
143 *fig.7*
144
chapter_01_get_to_know_FN.adoc 1.1 Render on change UTF-8 AsciiDoc 5 master +1, -1 2
```



I worked side by side with backend engineers to design a smooth User Experience for the content generation process and a seamless conversion phase operated by the toolchain.

> InDesign, AsciiDoc, AsciiDoctor, PrinceXML, HTML5, CSS3





ARDUINO REFERENCE – Publishing platform

With the experience gained building the Arduino Manuzio system we created a distributed digital publishing platform for the Arduino Reference.

Based on Github the system allows the **Arduino community to contribute to the documentation and translation of the Arduino Reference.**

One repository serves as the toolchain to convert the AsciiDoc content into HTML and PDF files adding correct styling and assets according to the destination of use. A repository for any of the different translation languages will store the user generated content in the AsciiDoc format.

I took care of the UX of the content generation process and co-designed and developed the UI of the reference pages for different destinations of use: Arduino website, Arduino IDE, Arduino CREATE, PDF books.

I created guidelines and example files to help the content generation process and worked side by side with the backend engineer to define a user friendly approach in the repositories structure.

> AsciiDoc, AsciiDoctor, PrinceXML, HTML5, CSS3

```

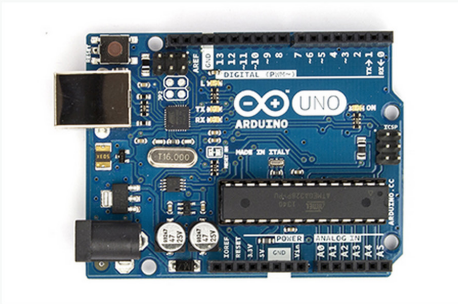
analogWrite.adoc
1 :source-highlighter: pygments
2 :pygments-style: arduino
3 :ext-relative: adoc
4
5
6 = analogWrite()
7
8
9 // OVERVIEW SECTION STARTS
10 [#overview]
11 --
12
13 [float]
14 == Description
15 Writes an analog value (http://arduino.cc/en/Tutorial/PWM)
16 [%hardbreaks]
17 On most Arduino boards (those with the ATmega168 or ATmega
18 The Arduino DUE supports `analogWrite()` on pins 2 through
19 You do not need to call `pinMode()` to set the pin as an d
20 The `analogWrite` function has nothing to do with the anal
21 [%hardbreaks]
22
23 [float]
24 == Syntax
25 `analogWrite(pin, value)`
26
27
28 [float]
29 == Parameters
30 `pin`: the pin to write to. Allowed data types: int. +
31 `value`: the duty cycle: between 0 (always off) and 255 (a
32
33
34 [float]
35 == Returns
36 Nothing
37
38
39 --
40 // OVERVIEW SECTION ENDS

```

analogWrite()

Description

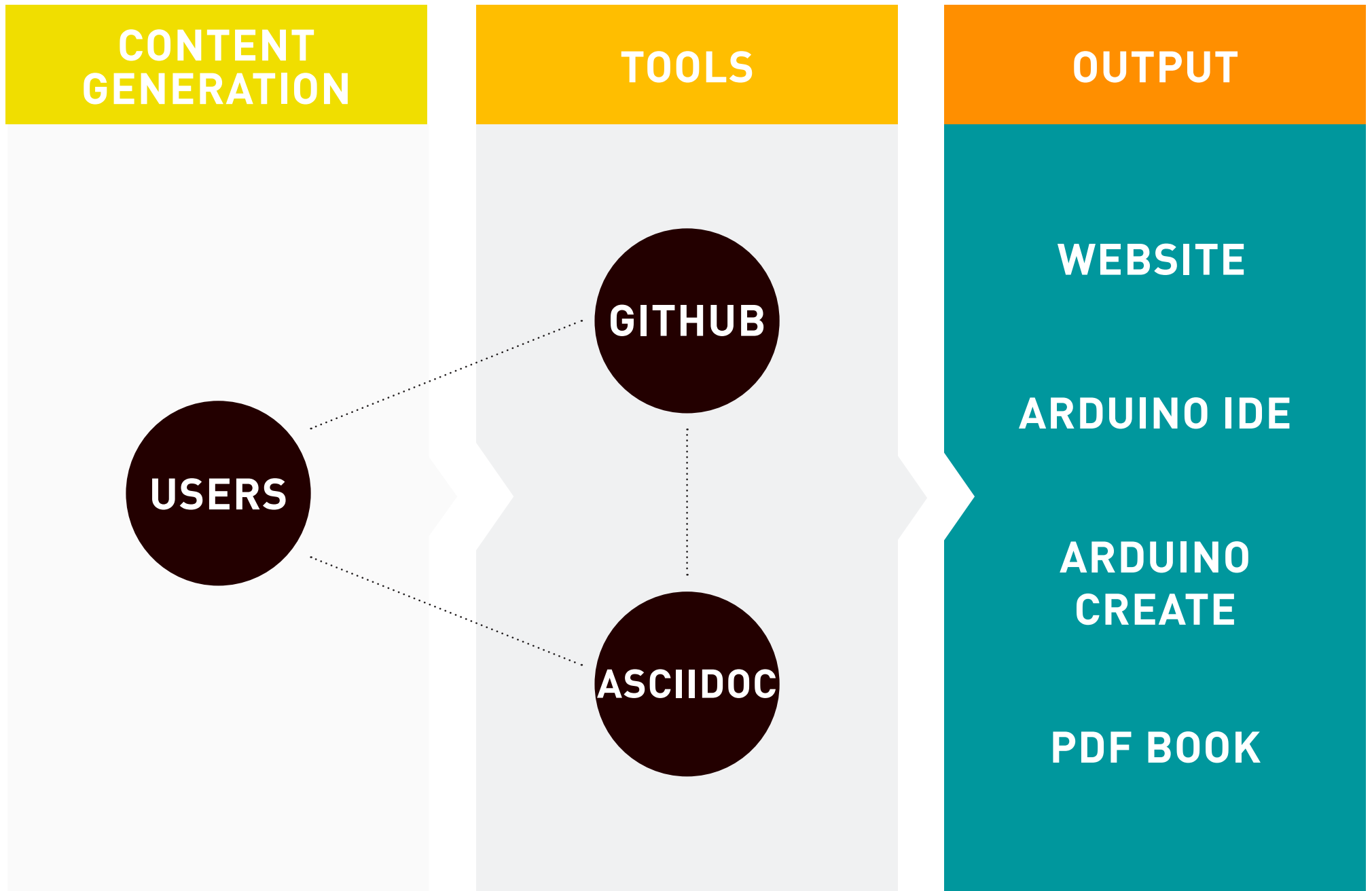
Writes an analog value ([PWM wave](#)) to a pin. Can be used to light a LED at varying brightnesses or drive a motor at various speeds. After a call to `analogWrite()`, the pin will generate a steady square wave of the specified duty cycle until the next call to `analogWrite()` (or a call to `digitalRead()` or `digitalWrite()` on the same pin). The frequency of the PWM signal on most pins is approximately 490 Hz. On the Uno and similar boards, pins 5 and 6 have a frequency of approximately 980 Hz.



Syntax

```
analogWrite(pin, value)
```





Folder Structure

```
tree -L 3 -I src
.
├── README.adoc
├── asciidoc
│   └── en
│       ├── AsciiDoc_sample
│       ├── LICENSE.md
│       ├── Language
│       └── README.md
├── configs
│   ├── javaide.json
│   ├── webide.json
│   └── website.json
├── templates
│   ├── css
│   │   ├── fonts
│   │   ├── style-font-base64.css
│   │   └── style.css
│   ├── index.html
│   ├── index.tpl.css
│   ├── index.tpl.html
│   └── sortedFolderStructure.json
```

Folder Structure

```
reference-en
├── AsciiDoc_sample
│   ├── AsciiDoc_Dictionary
│   │   ├── AsciiDoc_Template-Dictionary.adoc
│   │   └── attachments
│   └── Reference_Terms
│       ├── AsciiDoc_Template-Parent_Of_Entities.adoc
│       ├── AsciiDoc_Template-Single_Entity.adoc
│       └── attachments
├── Language
│   ├── Functions
│   ├── Structure
│   └── Variables
├── LICENCE.md
└── README.md
```

HOW.DO

HOW.DO – iOS App

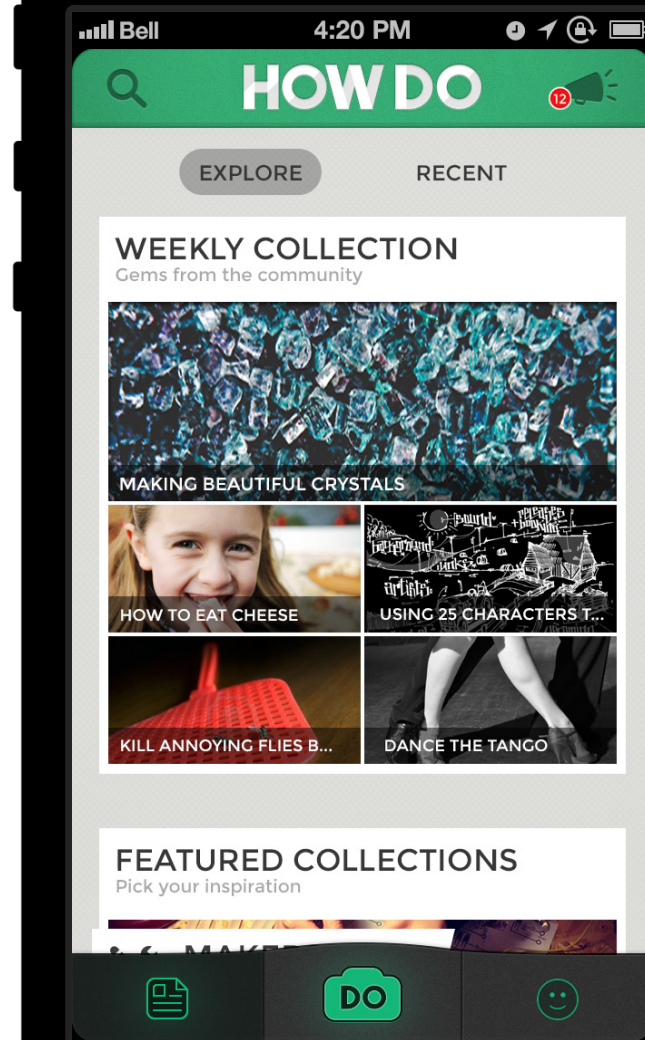
While working as Interaction Designer at **How.Do** I developed the UX and UI of the iOS app.

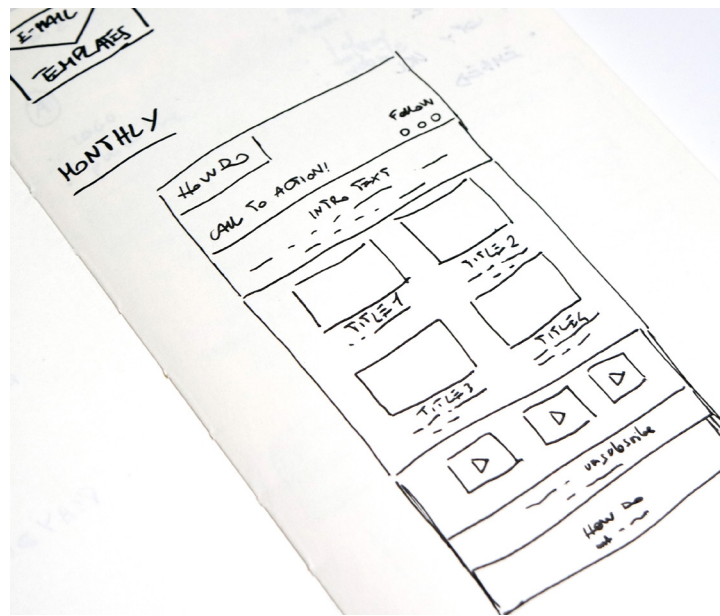
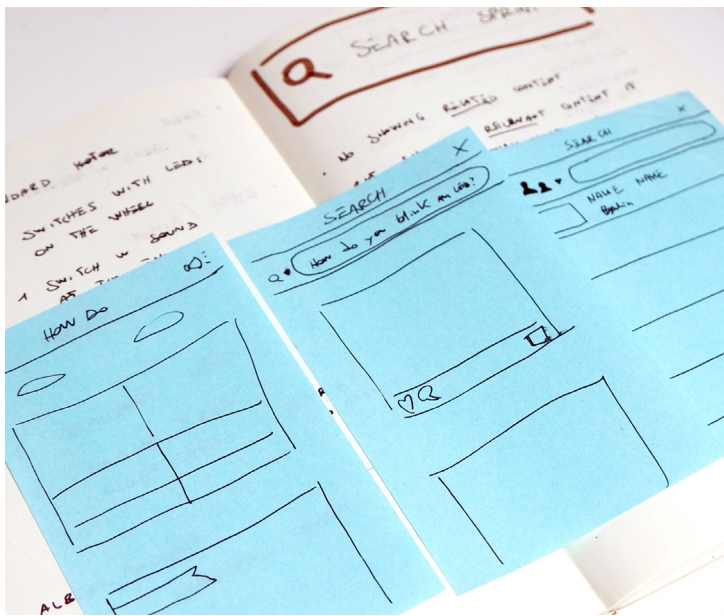
How.Do is a **platform to create and share Micro Guides and Tutorials** in the form of slide-shows of pictures and short clips with added audio commentary. It is also a social network based on learning and sharing learning experiences.

While working on the UX of the platform I **created userflows, mockups, wireframes** for all the different features and views of the mobile app.

← UI and Visual design wise I **designed and delivered production ready interfaces**: prototypes for all the possible screen resolutions, exports of assets for all possible hardware specifications (standard display, retina display)

> Photoshop, Illustrator, Paper and video prototyping

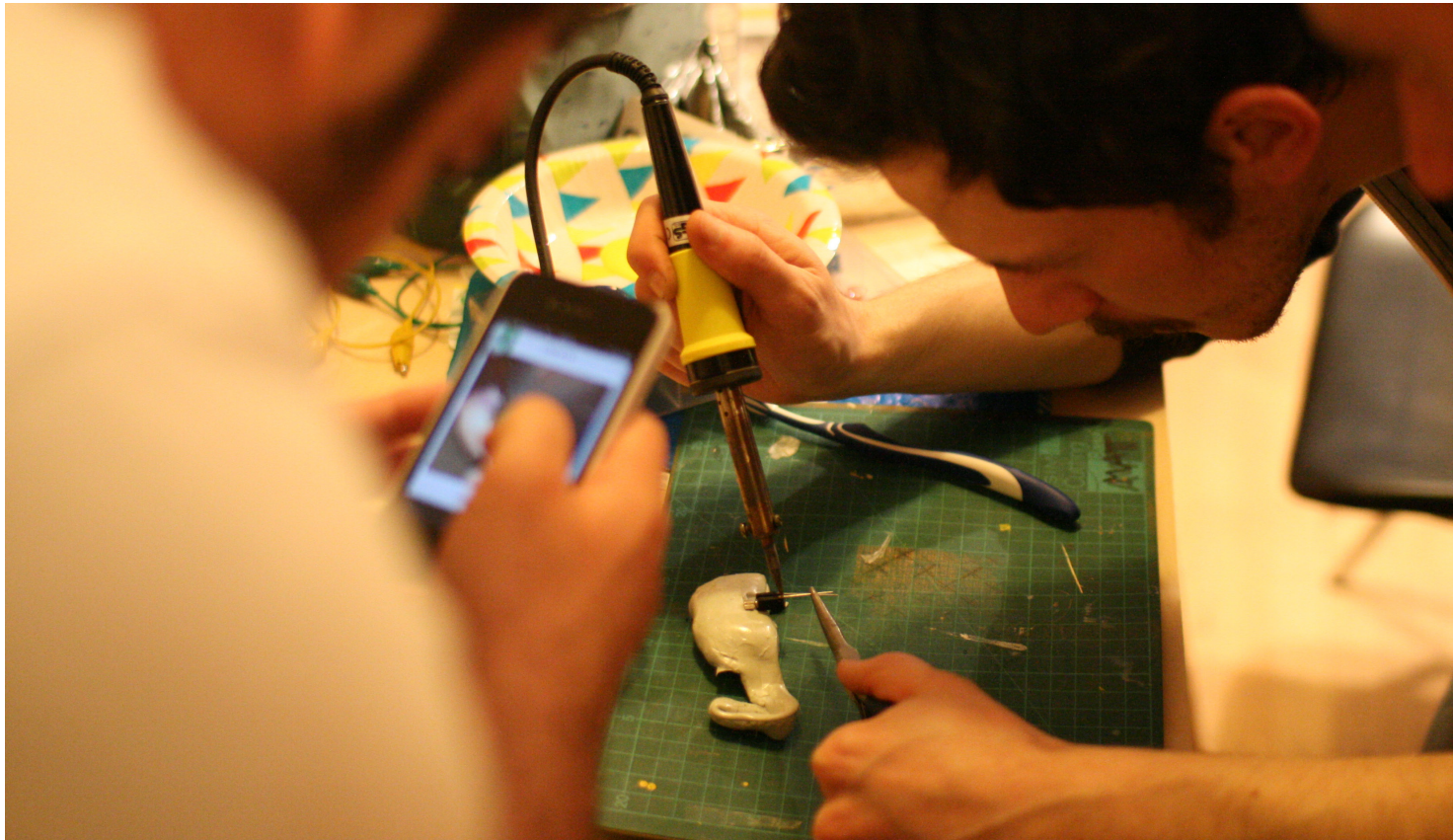




HOW.DO – Sketching and testing

← My approach in **designing and prototyping** screens for the How.Do app was to always go **paper-first**.

Whenever we had to cook a new feature I used to create **smartphone cutouts**, place them on top of screen sketches and ask colleagues and users to quickly test them.



← At How.Do we used to iterate very quickly on features, and we cultivated a culture of rapid prototyping.

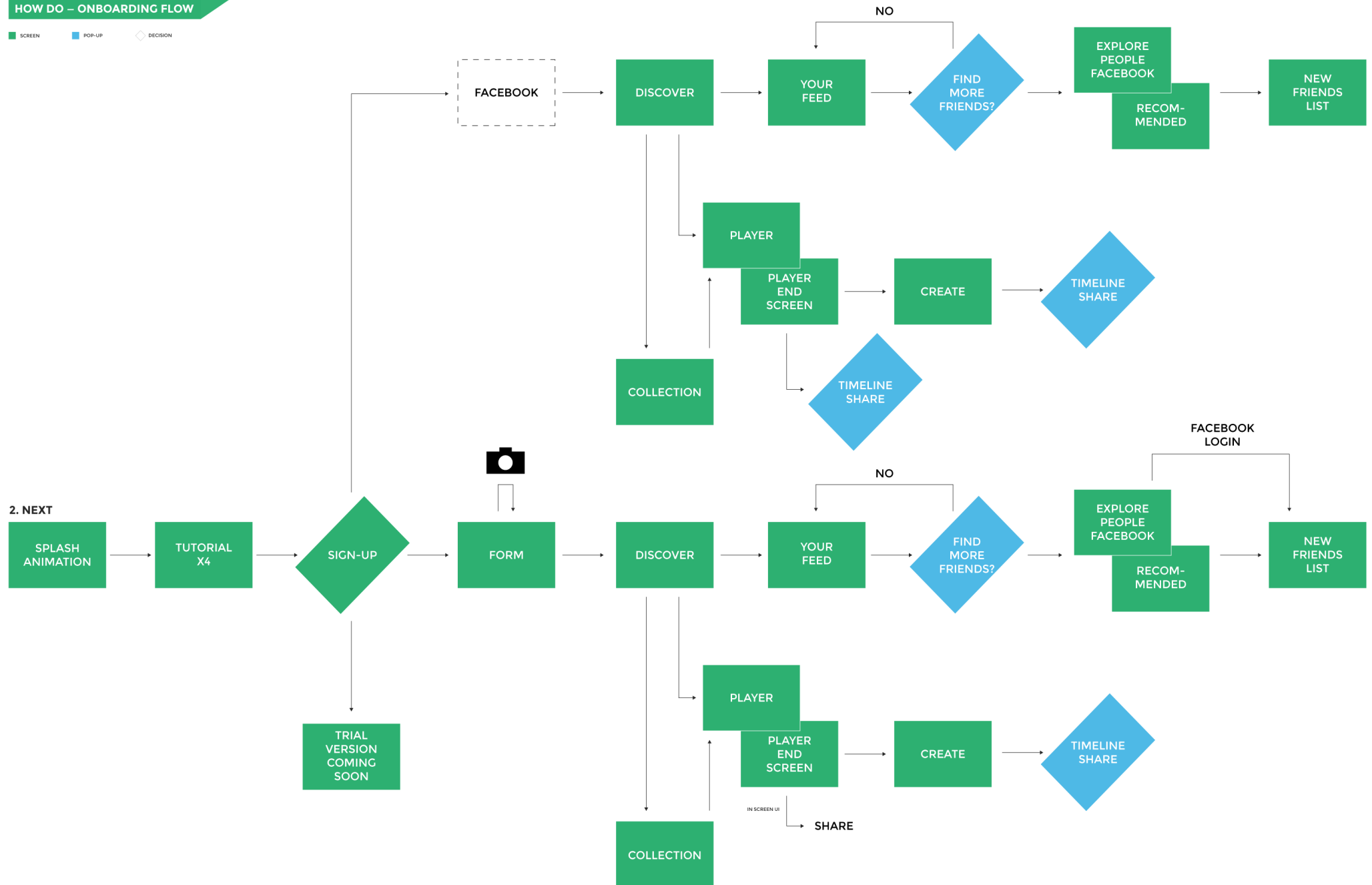
Testing was consistent in both the digital and physical space.

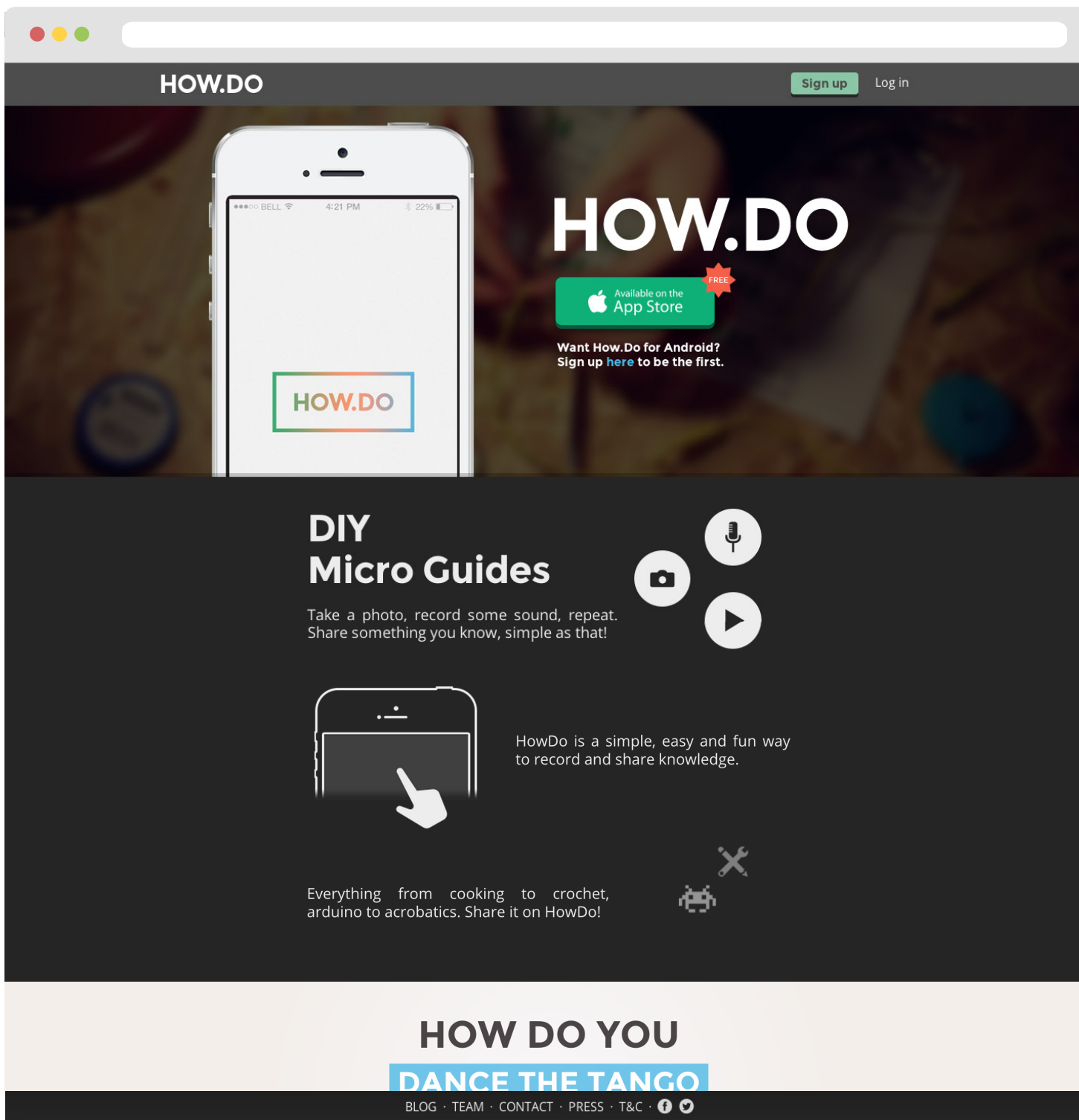
We were inviting makers and DIYers to co-creation sessions in our space and observed the tool in action.

Also, we explored online testing services such as Lookback and User Testing with mixed results.

HOW DO – ONBOARDING FLOW

■ SCREEN ■ POP-UP ◇ DECISION





HOW.DO – Web App

The How.Do web app is basically a browser oriented re-design of the iOS app.

It includes:

- Company micro website
- User profiles
- Micro guides' Web player
- Player embed

I designed the UX and UI of all the sections and took care of the UI Frontend development.

The web app is responsive and retina ready.

> HTML5, CSS3, JQuery

HOW DO

THIS MONTH'S PICK
POWER RELAX, HOW DO YOU DO IT?



DECEMBER 18TH 2012
INTO THE WILD

Hi Dirk, this weeks collection is all about jumping into the wild, and coming back in one piece!



SEARCH FOR
THE WHALE
BY IAN BACH



SEARCH FOR
THE WHALE
Readymade cardigan narwhal,
next level authentic cliché
high life before they sold.

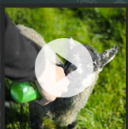


SEARCH FOR
THE WHALE
BY IAN BACH

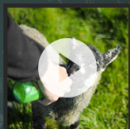


SEARCH FOR
THE WHALE
Readymade cardigan narwhal,
next level authentic cliché
high life before they sold.

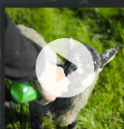
MOST TALKED HOWDOS



HOW DO YOU
PEEL A BANANA?



PIMP SNEAKERS
WITH SUGRU



TAKE OFF
PERMANENT

If you are reading from your phone:

TELL A STORY

You're subscribed to this inspiration by using the [HowDo app](#).
You can unsubscribe anytime.

HOW DO
HowDo. Made in Berlin.

HOW DO

1.1 IS OUT

NEW DESIGN

EASIER TO LOVE

SMOOTHER CREATION

We believe sharing knowledge is powerful. Showing somebody how to do something is one of the most beautiful human interactions we create. Last week we launch the first big update since release. Thanks for all the support and feedback along the way. We've listened and improved, this update is for you.

* We've simplified the feed to make it easier to discover interesting content.

* Brand new feed design that we've put lots of energy and love in.

* Curated feeds that shows collection of Maker/DIY, Food geekery, Life hacks and fun/quirky.

* It's easier to love content and create your own collection of what you like.

You're subscribed to this inspiration by using the [HowDo app](#).
You can unsubscribe anytime.

HOW DO
HowDo. Made in Berlin.

HOW.DO – Newsletter

While at How.Do I also **designed and wrote the code for a set of email templates** for transitional emails and newsletter campaigns:

- Monthly newsletter
- App release
- Community events
- Community contests

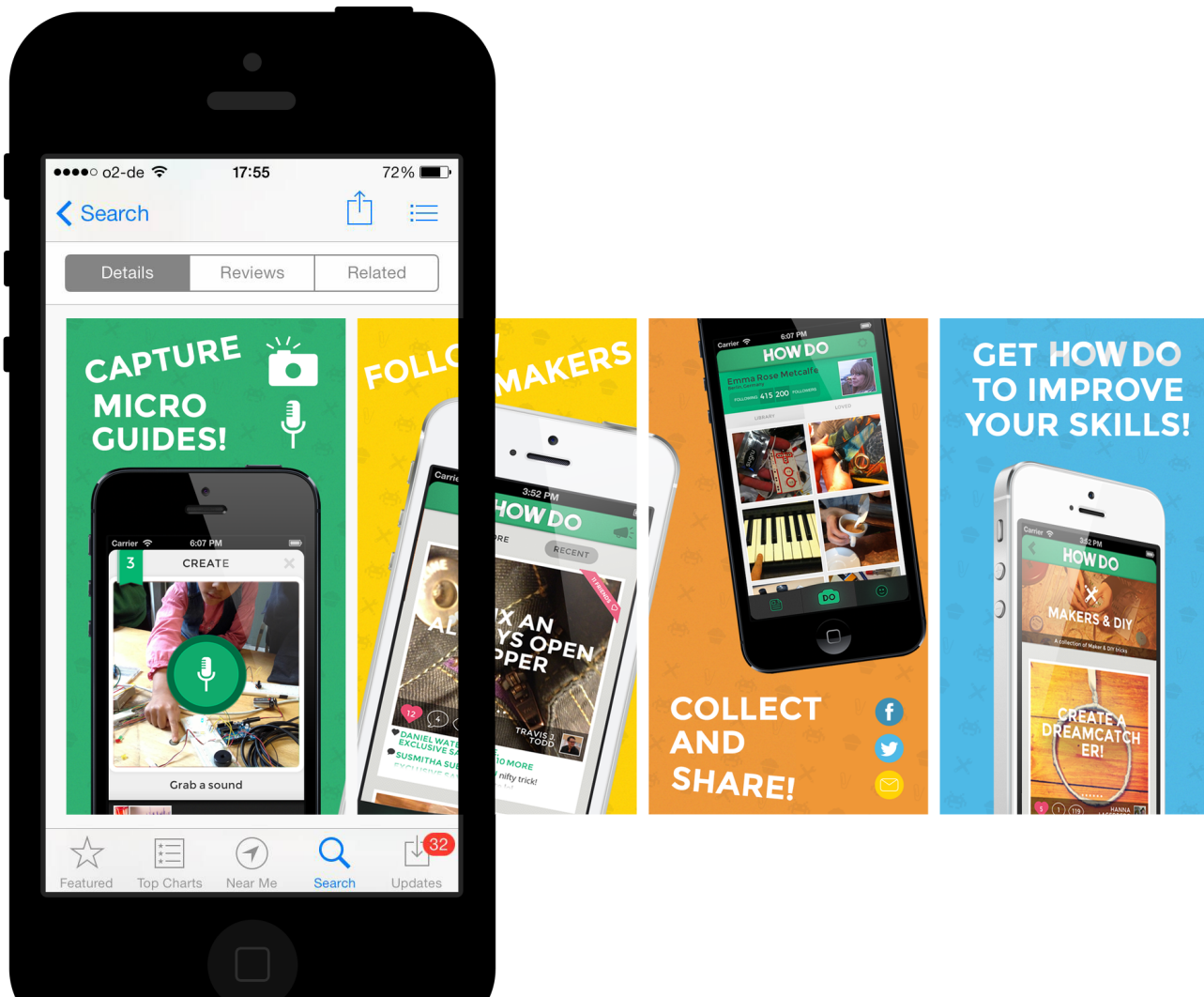
I designed the UI and took care of the UI Frontend development of the templates. The email templates are responsive and have been tested on all major email clients on desktop and mobile devices.

HOW.DO – Branding

While at How.Do I was also **in charge of the digital and physical branding** of the company.

For a series of events and community campaigns I designed:

- Business cards
- Stickers
- Flyers
- Banners
- Web banners
- AppStore ads



EMMETT ARCHIVE – NY Public Library

While attending the **Hackership** program I was contacted by the Information Design firm **ACCURAT** to develop a research project for the **New York Public Library**.

The interactive data visualization project collects the vast majority of the correspondence of the Founding Fathers during the days of the Declaration of Independence in the USA.

I developed a micro website featuring a data driven visualization of the collection of writings while experimenting and learning about D3.js.

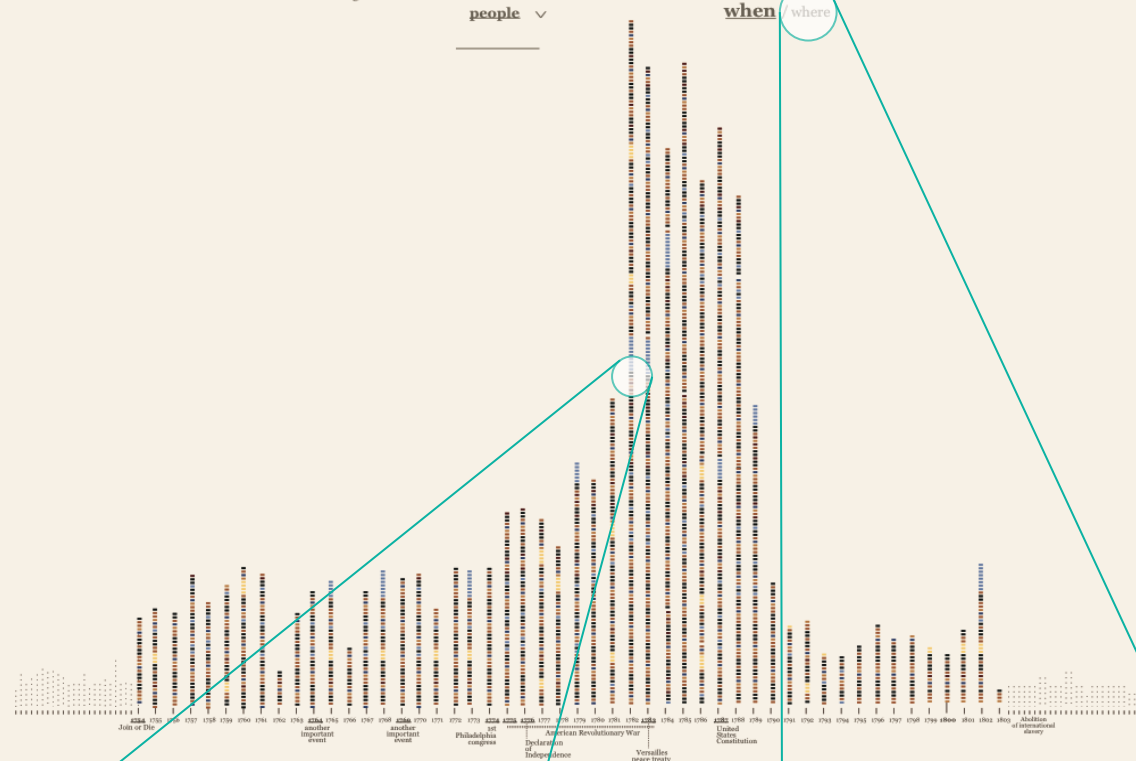
- > AngularJS: Single Page App structure, Search feature
- > D3.js: Translation of the Json dataset into a dataviz
- > SVG, CSS3, jQuery, Bower, Github

Founding fathers' correspondence

Visualizing the Emmett collection

people ▾

when / where



from Washington (George)
to Amanda Pikin
1781, September 19

place
Hartford, Conn.

original chapter

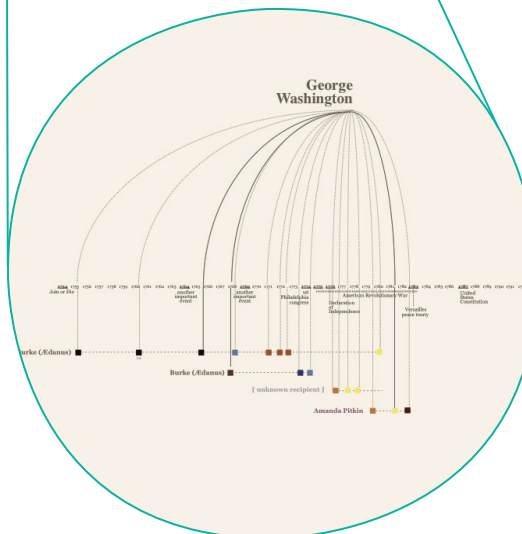
The Albany Congress of 1754

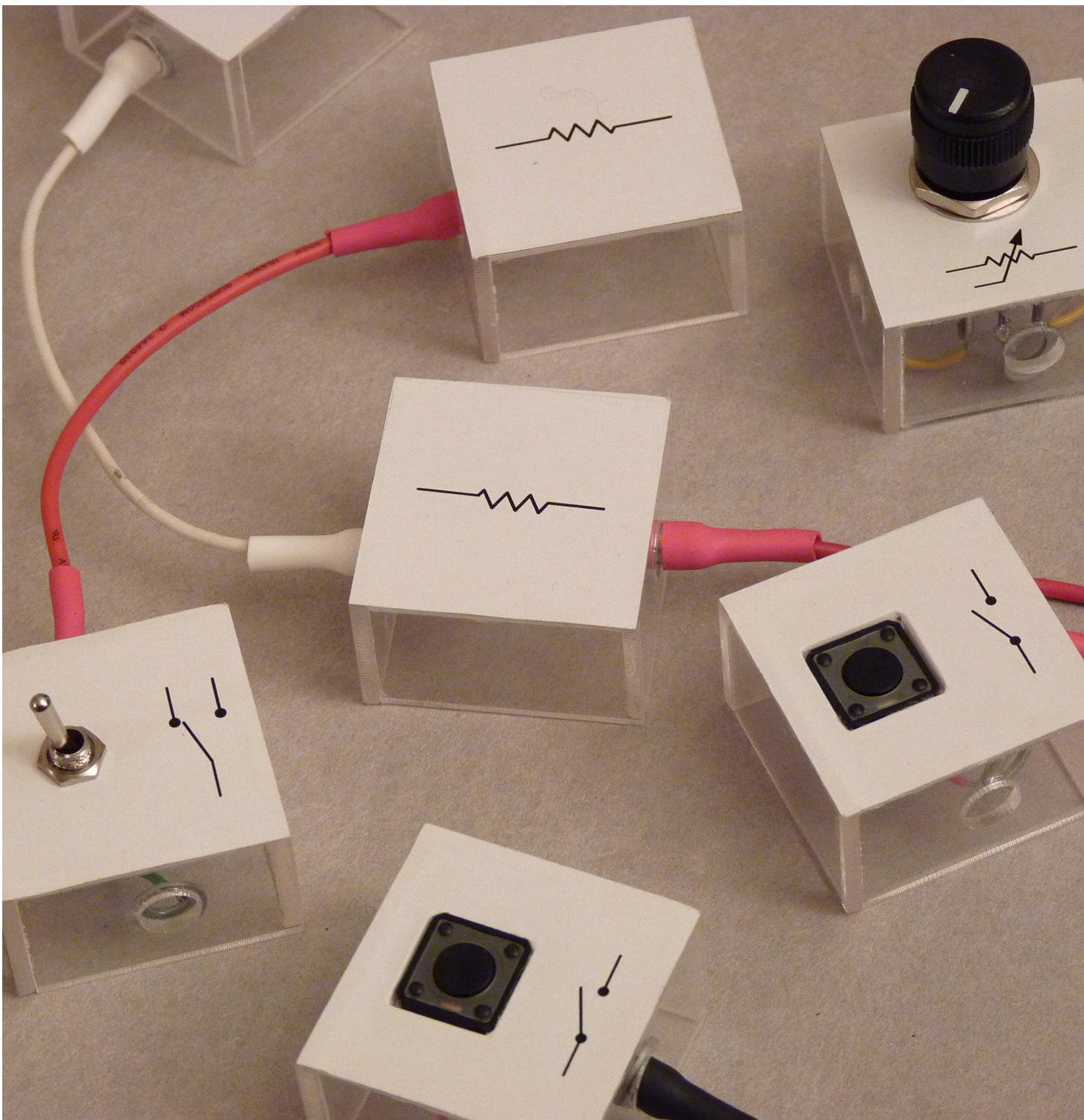
scope
Recruits for the army ; Capt. Burnham of Middleton ; company raising to be commanded by Capt. Stephen Lee, and Lts. Prentiss and Dow ; hears that the Governor of New York has gone to Albany

entry
Wyllys (George). Hartford, [Conn...]
Sept. 19, 1755. To William Pitkin.
Recruits for the army ; Capt. Burnham of Middleton ; company raising to be commanded by Capt. Stephen Lee, and Lts. Prentiss and Dow ; hears that the Governor of New York has gone to Albany. A. L. S. 2 pp. Fo

[go to original document](#)

George Washington





SYNTHUINO

Synthuino is my graduation Thesis project for the **BSc in Computer Science and Music Information Technology** at the University of Milan.

Synthuino is a **modular hardware platform designed to teach the basics of electronic sound synthesis**.

Plexi bricks containing electronic components can be assembled and magnetically snapped together, and to the main hub containing an Arduino.

Following a set of templates the user can build actual audio synthesizers.

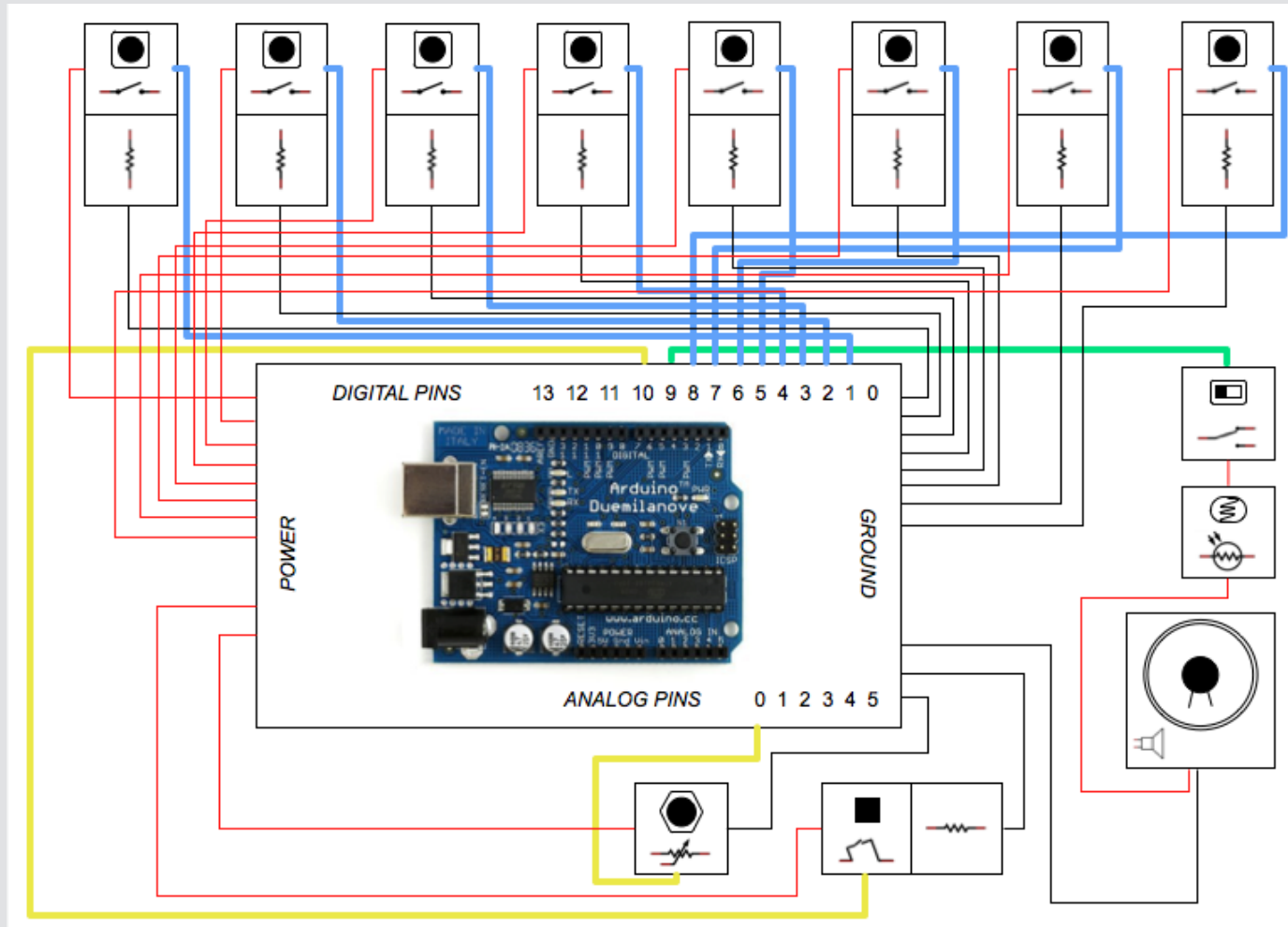
Synthuino makes it possible to create and manipulate sounds and to experiment a variety of physical interactions with the synthesizer.

The project was designed and developed in 2010.

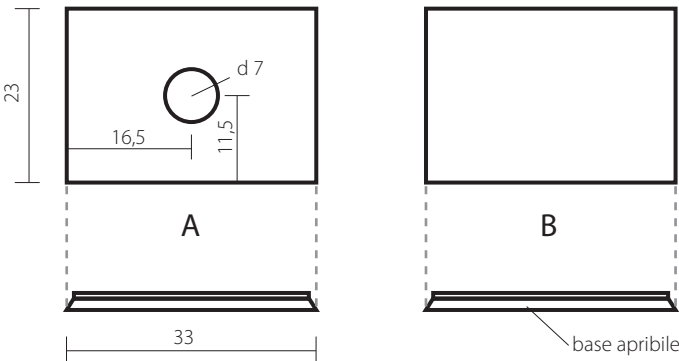
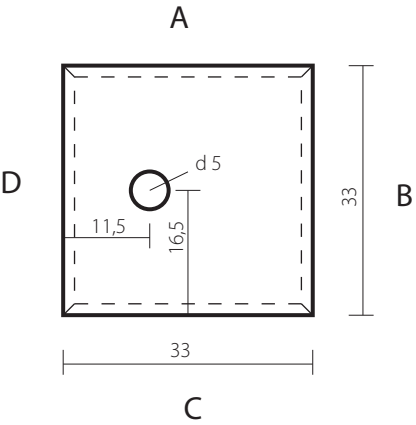
http://bit.ly/alessandro_contini_thesis_ITA

Keyboard synth

circuit diagram



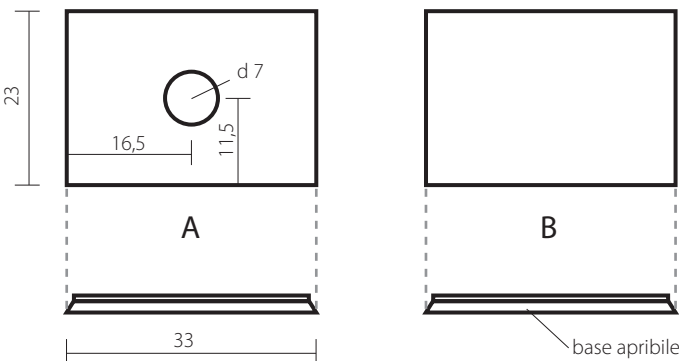
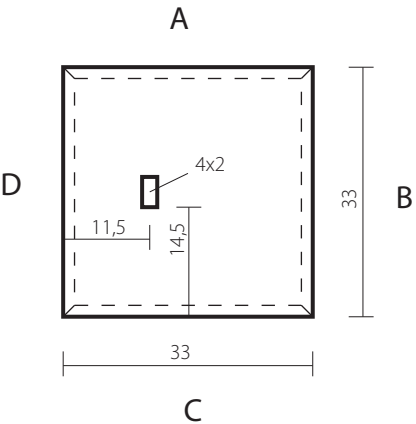
interruttore x 2 - metacrilato trasparente spessore 1,5 mm



VISTA DALL'ALTO

VISTE FRONTALI

fotoresistore - metacrilato trasparente spessore 1,5 mm



VISTA DALL'ALTO

VISTE FRONTALI

THANK YOU

CONTACTS

web: www.alessandrocontini.it
email: info@alessandrocontini.it
skype: alessandro_contini
phone: +34 666 182 949

linkedin: www.linkedin.com/in/continialessandro/
github: www.github.com/cntlsn
twitter: [@CNTLSN](https://twitter.com/CNTLSN)