Accepted Manuscript

Hardware Article

Shybo. An open-source low-anthropomorphic robot for children

Maria Luce Lupetti

PII: S2468-0672(17)30038-X
DOI: http://dx.doi.org/10.1016/j.ohx.2017.08.003
Reference: OHX 12

To appear in: *HardwareX*

Received Date: 4 August 2017
Revised Date: 29 August 2017
Accepted Date: 29 August 2017

Please cite this article as: M.L. Lupetti, Shybo. An open-source low-anthropomorphic robot for children, *HardwareX* (2017), doi: http://dx.doi.org/10.1016/j.ohx.2017.08.003

HardwareX article template

[Please fill in the template below and delete all instruction text in italics before submitting.]

GENERAL INFORMATION
HardwareX is an open access journal established to promote free and open source designing, building and customizing of scientific infrastructure (hardware). For more details on best practices for sharing open hardware see http://www.oshwa.org/sharing-best-practices/

Title: *Shybo. An open-source low-anthropomorphic robot for children.*

Authors: *Maria Luce Lupetti*
Affiliations: *DAD, Department of Architecture and Design, Politecnico di Torino*
Contact email: *maria.lupetti@polito.it*

Abstract: This article presents Shybo: a novel low-anthropomorphic robot for children. The robot, resulted from the combination of open-source hardware and software, is able to perceive sounds and to react through two non-verbal behaviors: hat's movement and lighting. By taking advantage of an open-source machine-learning software, the robot can be easily trained by children. This robot can be employed in research to support human-robot interaction studies with children, for investigating perceptual aspects of robot's features or for investigating children' cognitive abilities. It can also be used for applications in educational context to support playful learning experiences.

Keywords: *child-robot interaction; open hardware; open design; 3D printing; machine-learning;*

Specifications

Hardware name	*Shybo*
Subject area	Educational Tools and Open Source Alternatives to Existing Infrastructure
Hardware type	*Other: Sound-reactive learning robot.*
Open Source License	*GNU General Public License (GPL) 3.0*
Cost of Hardware	*80 $ - 171 $*
Source File Repository	*https://osf.io/xk2r4/#*

---

1. Robot for child-robot interaction studies and playful learning applications

In the field of human robot interaction (HRI) studies with children, the use of low-anthropomorphic robots, able to perform minimal behaviors, is popular. These are employed with different purposes that vary in terms of the focus of investigation, determining the role of the robot. In soma cases, in fact, the studies are aimed at investigating the role and efficacy of certain robot's behaviors, and how these are perceived from people. In these cases, the robot is protagonist of the study.
For instance, Fink et al. [2014] designed, developed and tested the robotic box "Ranger" for understanding which robot behavior can motivate children to tidy up their toys. According to the purpose, the robot was designed as a box able to move around and to communicate basic facial expressions through eyebrow. It was also able to give a feedback to children who put a toy on it, by emitting minimal sound and light cues. Similarly, Zaga et al. [2017] designed a low-anthropomorphic robot able to perform a minimal movement, namely a gaze movement, for investigating the effect of such a minimal behavior on children's perception of robot's animacy and likeability. Another example is represented by the Haptic Creature [Yohanan and MacLean, 2011], and the related study carried out by Yohanan and MacLean [2011]. In this study, the authors investigated the effectiveness of nine different behaviors, obtained through ears position, breathing rate and purr, for the communication of emotional states.
Other studies, instead, employ low-anthropomorphic robots as mediators for investigating children's cognitive abilities and developmental stages. Michaud et al. [2005], for instance, designed a spherical robot called Roball with the aim of studying different elements of child development theories. A similar kind of robot was used by Boccanfuso et al. [2016]. In their study, in fact, they used the Sphero robot (http://www.sphero.com/sphero), a commercially available robotic ball that can be teleoperated through a mobile app. This robot can move around, change speed and light up in different colors and fade. Boccanfuso et al. used Sphero as an emotion-simulating robot to conduct studies with children affected by Autism Spectrum Disorder. Another example is provided by the study by Kozima et al. [2009], who though used a very different robot: Keepon. This small social robot, despite it cannot move in the environment, is characterized by minimal body movements that enable it to interact socially with children. The authors, in fact, used Keepon to conduct studies with children for investigating various levels of physical, mental and social development.
Some of the robot mentioned, together with a variety of other commercial robots, are also starting to be widely employed also for daily-life applications, especially in educational contexts. These commonly are used as tools for playful learning experiences, often focused on teaching coding to kinds. These robots can be used to support STEAM education, such as in the case of Sphero, which was employed in a mobile computing and robotics class [Kurkovsky, 2013]. Certainly, the most popular robot for educational applications is the LEGO Mindstorm, which is widely adopted to teach STEAM subjects [Benitti E Spolaor, 2017], support storytelling activities [Ribeiro et al., 2009] and foster problem solving abilities [Hussain et al., 2006].

2. The potential benefits of the open-source approach

In both research and applications, the design and release of low-anthropomorphic robot as open source project can bring significant benefits.

Regarding research, it can firstly result is a *reduction of costs*. The use of open-source platforms, in fact, is getting increasingly common in studies that employ robotics hands for leveraging low cost technologies such as 3D printing and open-hardware, e.g. microcontroller-based boards [Bulgarelli et al., 2016]. A second benefit consists of the possibility of having full *control on the robot's features*. As a matter of fact, certain robot's features, such as appearance, locomotion mode, gender, and personality, can greatly affect people feelings and attitude toward a robot [Woods, 2006]. A third motivation for employing open-source platforms in HRI is the *replicability of studies*. Many investigations, in fact, are conducted with custom hardware, which, in most of cases remains for the exclusive use of the authors. However, as explained by Baxter et al. [Baxter et al., 2016], the possibility of conducting the same experiment anew is functional for weaving a solid and scientific fabric, but unfortunately there is usually lack of replicability in HRI studies.

Also regarding applications for daily life contexts, such as schools, the adoption of open-source platforms can bring a *reduction of costs*. Gonzalez-Gomez et al. [2012], for instance, designed 3D printable and open-source mobile robot that can be employed in schools for educational activities, that requires a very limited cost: 57 €. The possibility of reducing costs is particularly relevant in school contexts were the activities are carried out with numerous groups of children and the costs multiply. A second benefit for educational applications is the possibility of *customization*. By adopting open-source platforms, in fact, the robot can be potentially modified according to the purposes of the designed experience. For instance, non-verbal behaviors or physical features can be changed to match it with a specific storytelling, enabling stronger connections between the new activities proposed and the existing curricula. Finally, by adopting and releasing open-source platforms, schools and other educational entities might benefit from the *increase of educational resources* freely available, meeting the philosophy of the open education movement [Atkins et al., 2007].

By referring to these motivations, a novel low-anthropomorphic robot for children was designed and developed. The robot was results from the combination of open-source hardware and software and it is released under a GNU General Public License (GPL) 3.0. The complete documentation is available at [Lupetti, 2017].

3. Shybo

Shybo is a low-anthropomorphic robot that perceive sounds and react through two minimal behaviors: hat's movement and light, that change fade and color. The design of the robot was based on preliminary studies conducted with parents and children [Lupetti et al., 2017 A, Lupetti et al., 2017 B].

The robot can show two main emotional states: active and scared. In the active status, the hat is open and it changes color according to sounds, while in the scared status it closes the hat, lights up in red and shakes.

A peculiar aspect is that it doesn't have preset color-sound associations and it can be trained by children every time. The face of the robot, in fact, consists of its interface: the mouth is a button, the nose is a potentiometer and the eye is a LED ring. Each of these elements communicate with a machine learning software. By turning the potentiometer, it is possible to select a color category, which is visible on the LED ring. Then, after choosing a color, it is possible to associate a sound by pushing the button, and the robot starts recording a new audio sample. In this way children can decide their own association and play with the robot by making sounds with different objects: from musical instruments to daily life objects. The two main functioning modalities, play and train, can be activated by moving a switch that is located on the bottom of the robot.

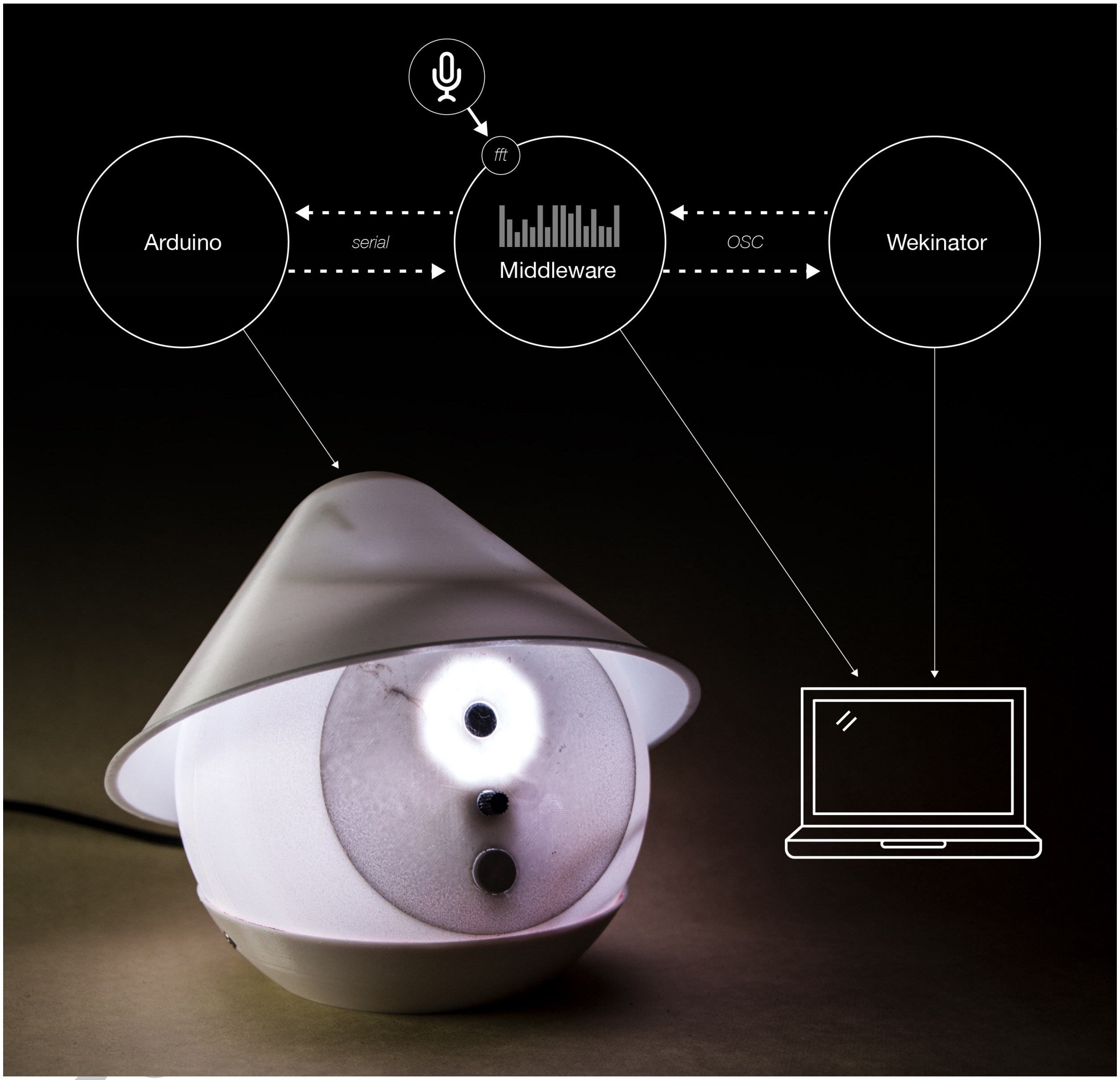


Fig. 1. The Shybo's prototype and the schema of the current architecture: the hardware is supplemented with a computer to perform the sound analysis and machine learning. The robot runs an Arduino sketch that controls the behaviours.

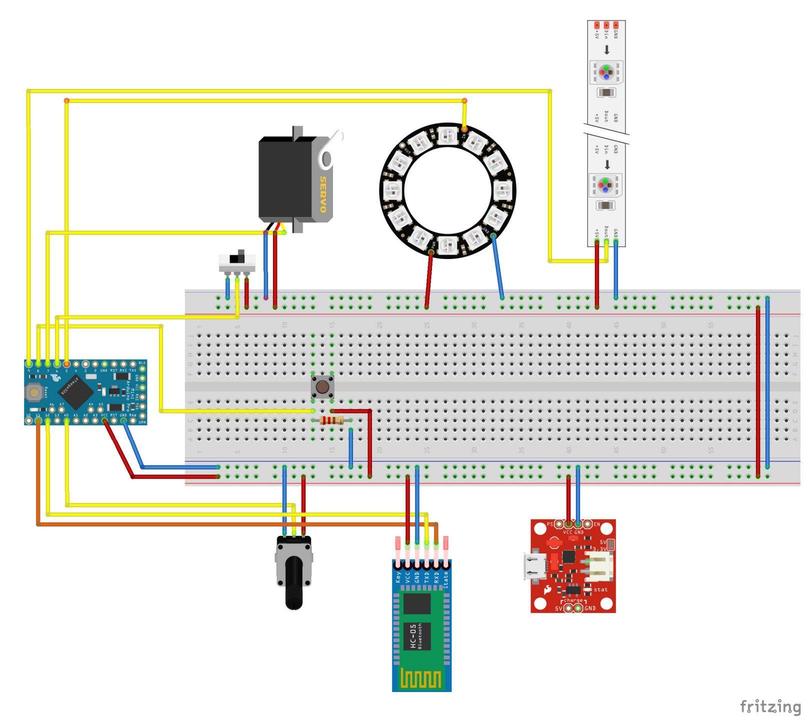
Shybo's functioning is obtained by combining open source software: Arduino, Processing and Wekinator (see figure 1). The Arduino runs a sketch that control the robot's behaviors and communicate via Bluetooth with a laptop used to run the sound-analysis middleware, designed in Processing by Romagnoli [2017] and the machine learning software for classification, namely Wekinator [Fiebrink et al., 2009]. The robot needs also to be physically connected to the laptop via USB, to get power. The use of the laptop is due to the fact that Shybo does not include a functioning microphone and a battery yet, and the machine learning software requires a Raspberry Pi board to be embedded in the robot.

Although, all these elements are not yet embedded, this setup allows to prototype, easily and fast, a Shybo robot suitable for studies with users and educational applications.

2.1. Electronics

Shybo is composed by ten main electronics components. Figure 2 provide a simple schema of the connections required to build Shybo and the pin used to connect each component to the board. This schema was designed with *Fritzing*, an open-source initiative that provide a software useful for documenting and sharing prototypes, and manufacturing pcbs. It can be downloaded at http://fritzing.org/home/. This schema was designed with the intent of illustrating the correct connections of the various components. However, the final assembly of Shybo requires the use of a small Protoboard (size: 3x7 cm) where the components have to be connected on both sides, as shown in figure 3.

Shybo, in fact, was designed with the intent of obtaining a small size robot that can be easily handheld by a child. To do so, the assembly has to be compact. Nevertheless, it is recommended to build a preliminary prototype by using a breadboard as shown by Figure 2. This allows to easily test the functioning.



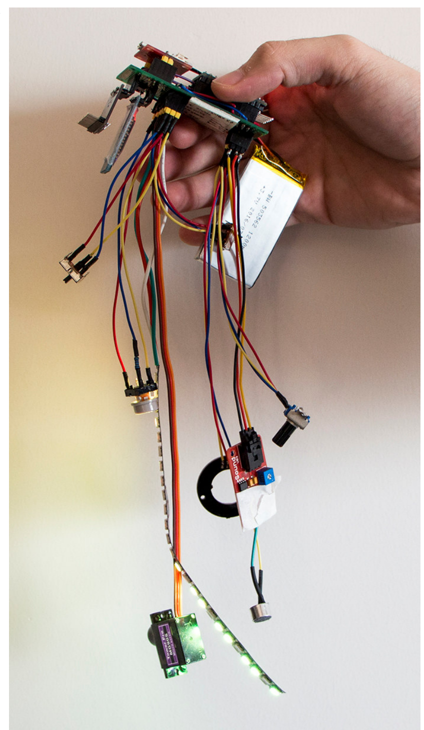


Fig. 2. Simplified schema of the electronics components necessary to build Shybo. **Fig. 3.** Actual schema of the electronics components necessary to build Shybo. In the figure there are also two additional components, such as touch sensor and a battery, that can be easily added.

2.2. Software

As previously mentioned, the Shybo's functioning requires the use of three software: Arduino, Processing and Wekinator. A complete folder, that includes the Arduino sketch, the Middleware and the Wekinator model, can be downloaded at https://osf.io/xk2r4/#.

The robot's behaviors were designed as a finite state machine, illustrated in figure 4.

The Arduino Ide is used to design the sketch that will control the robot and it can be downloaded at https://www.arduino.cc/en/main/software. The Processing software (that can be downloaded at https://processing.org/download/) is used to run the middleware, which has two roles. On one hand, it receives real-time sound data and performs a FFT (Fast Fourier Transform) splitting sounds in 250 bands. On the other hand, it exchanges data between the Arduino and the Wekinator, which is a free, open source software for machine-learning that allow to easily train and modify many standard machine learning algorithms in real-time (Fiebrink et al., 2009). The software can be downloaded at http://www.wekinator.org/downloads/.

Figure 5 illustrates the logic of the robot which was developed as a finite state machine, composed by six states. These states are associated to the two functioning modalities of the robot: play mode and train mode. The transition from one mode to the other is determined by the position of the left switch, located on the bottom surface of the robot.

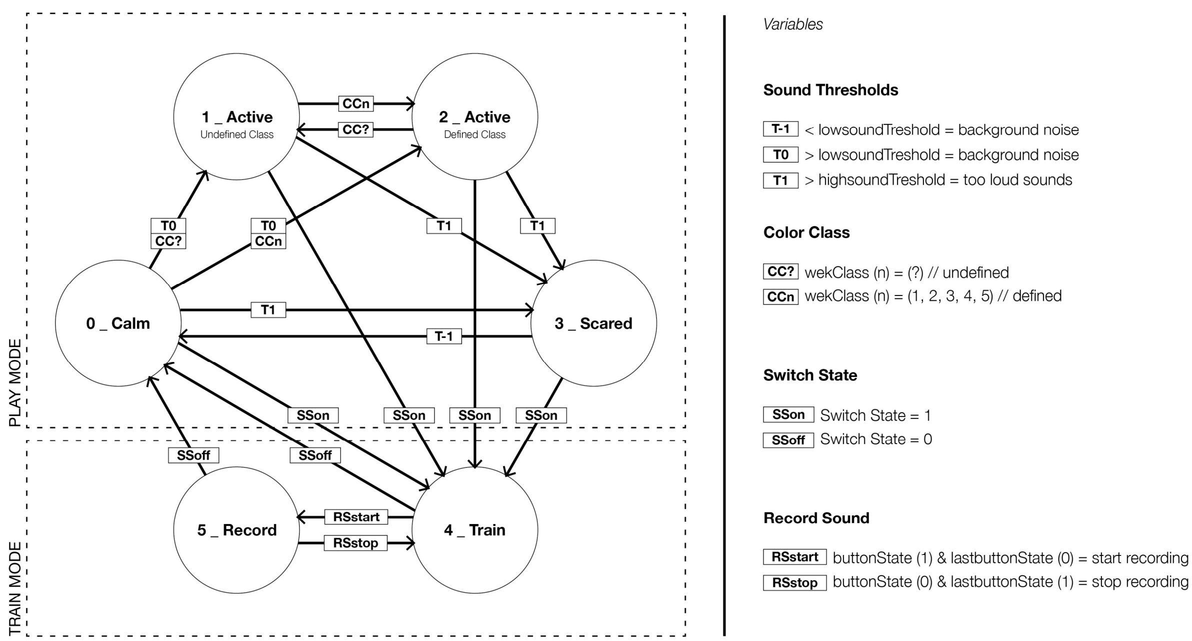


Fig. 4. Finite state machine that regulate the robot functioning.

In the *train mode*, the robot can send messages via Bluetooth to the middleware, which forward them to the Wekinator via OSC (Open Sound Control). It can send two kind of messages: color class and button state. By turning the potentiometer (the nose of the robot) it is possible to change the color class to which associate a new sound. The color class is visible on the body of the robot that lights up accordingly. When the button (the mouth of the robot) is pressed, instead, the status change from 4 to 5 and the Wekinator starts recording a new sound.

In the *play mode*, instead, the robot can change 4 states: calm (0), active with undefined class (1), active with defined class (2), and scared (3). All these states are determined by the loudness of the perceived sounds. The calm state is activated when the robot is switched on. If the sounds exceed the low-sound threshold the robot moves to the status 1 or 2, depending on if the robot was previously trained with the perceived sound. When the class is undefined the robot lights up in a sequence of colors rather than a single color. In status 2, instead, the robot lights up changing color according to the color-sound combinations trained to the robot. The recognition of the sounds and the activation of a color class is performed by the Wekinator, who receive sound data from the middleware.

2.3. Design

Shybo was designed with the primary intent of enabling a friendly interaction. This was pursued in two ways. On one hand, Shybo was designed to allow children to grab and hold it with their hands. For this reason, it has a rounded shape and a small size, showed by Figure 5. On the other hand, both the morphology and the behaviors were designed to avoid uncanniness, by adopting an iconic style and using a hat and lights to communicate the states, instead of alternatives such as facial expressions. The hat can assume three positions, by rotating by 15 or 30 degrees.

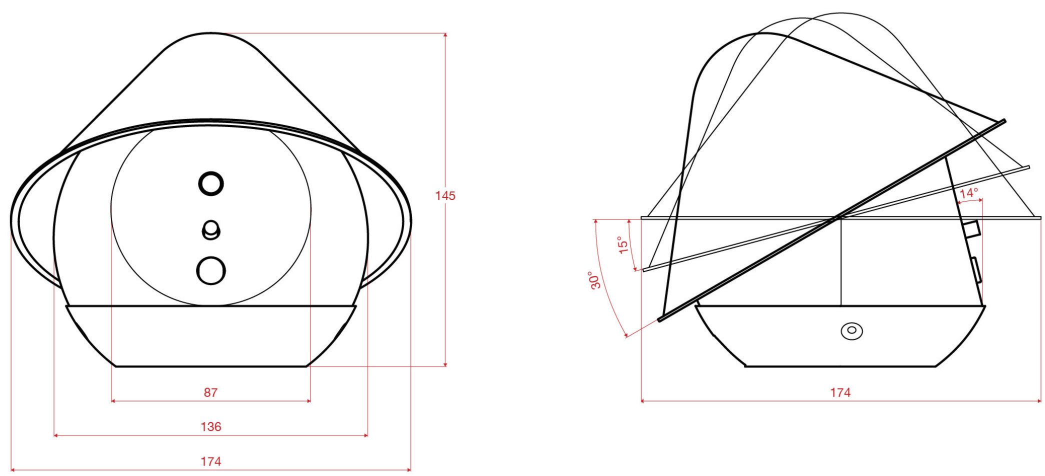


Fig. 5. Shybo's front and side view with main dimensions.

As shown by Figure 6, the body of the robot is composed by four main elements: hat, front shell, back shell, and base. The front shell is designed to lodge the button, the potentiometer and the LED ring. The current design has also a hole for a microphone, which however is not used in this model. This can be used for iterations or for mere aesthetical purpose, replacing the microphone with a 3D printed cylinder of the same size (the precise size can be seen in the 3D model). Other smaller elements, namely the axis and the servo docking, are used both to hold the electronics and to connect the main elements. Only the LED structure is an exception, since it is used just to hold the LED strip.

All the elements were 3D modelled using Rhino and 3D printed. The final model of each component is provided in stl format, while the entire model is provided in both 3dm and blend formats. All the components can be downloaded at https://osf.io/xk2r4/#. The elements were printed with an FDM 3D printer in white PLA, by a professional printing service. This choice was determined by time issues, however by printing in your own lab you should considerably reduce the costs of the prototype.

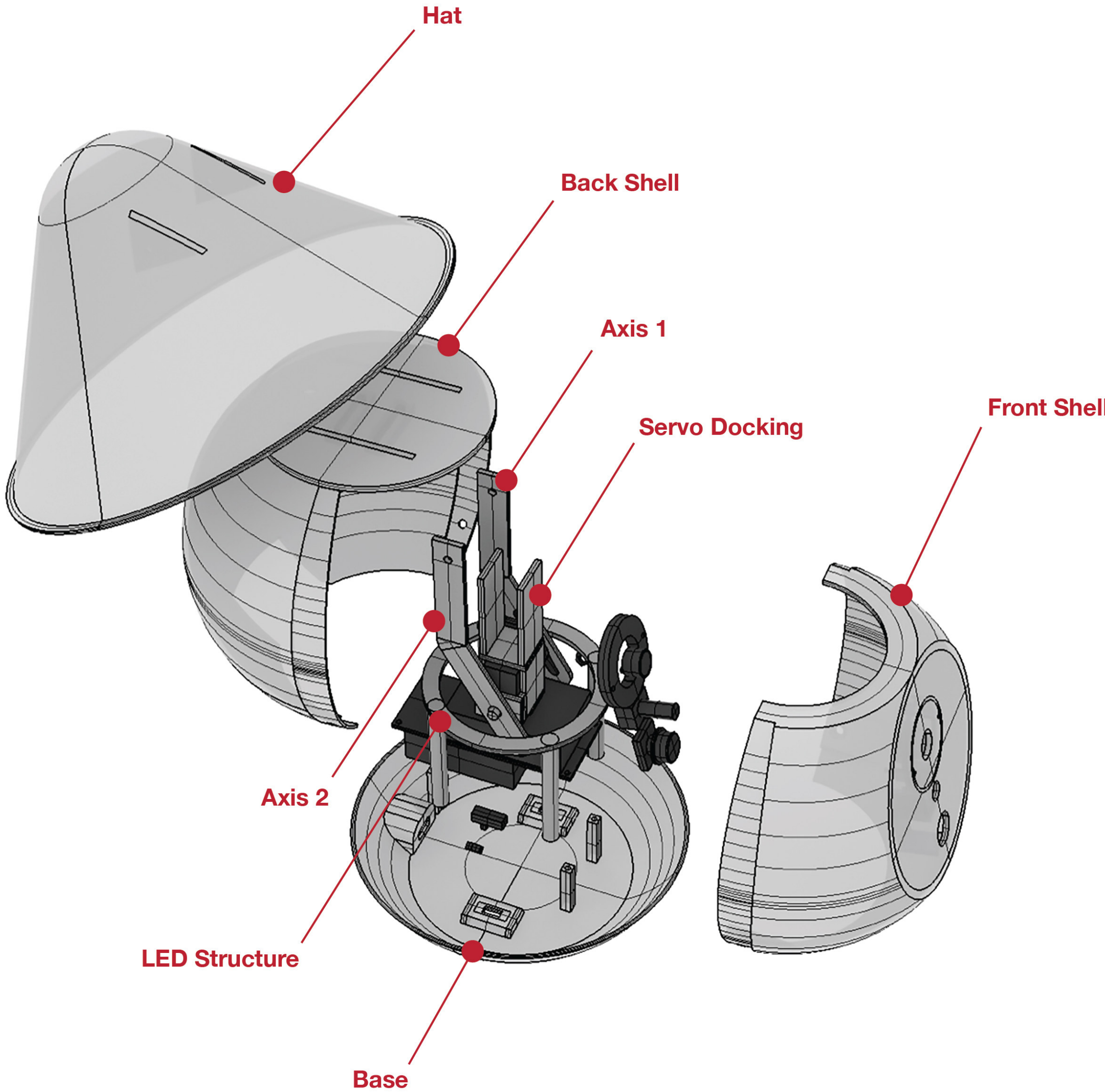


Fig. 6. Shybo's exploded view with main components.

Design Files Summary

Table 1. The table presents the list of the design files.

Design file name	File type	Open source license	Location of the file
Entire model	3dm; blend	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Hat	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Back Shell	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Front Shell	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Base	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Axis 1	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Axis 2	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
Servo Docking	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/
LED structure	Stl	GNU General Public License (GPL) 3.0	https://osf.io/x82yg/

In table 1 are listed all the design files necessary to build Shybo.

The *Entire Model* file consists of a complete model that include all the body and structure elements that need to be printed, together with the main electronics components. This allows to have a general view of the robot and of where the electronics is placed.

The *Hat* file is a hollow cone, rounded on top. It has to be printed from the bottom to the top using supports.

The *Back Shell* file consists of half of the robot's body. It has to be printed from the top to the bottom, using supports.

The *Front Shell* file is the second part of the robot's body. It has to be printed from the front, the face, with supports.

The *Base* file consists of the bottom part of the robot. it has to be printed from the bottom to the top, with supports.

Axis 1 and *Axis 2* files are the elements that connect the hat to the servo and the servo docking. They have to be printed on one side, without supports.

The *Servo Docking* file contains the element that hold the servo and has to be connected to the Back Shell. It has to be printed from the right side to the left, without supports.

The LED Structure file is a small rounded ring supported by four cylindrical columns. It has to be printed from the top to the bottom without supports.

4. Costs

The bill of materials, in table 2, provides the list of both electronics and design components necessary to build Shybo. The total cost of the prototype, according to the table, is 171 $, around 150 €. This number, however, is greater than the actual expenditure incurred. It represents the maximum cost required by the project, which can be significantly reduced.

First of all, the electronics listed in the table include the purchase of components that are often available in labs and that usually come in packs. The button, for instance, is included in a set of 15 pieces. If we split the cost of the set for each piece, the cost of this component goes down from 5,95 & to 0,4 $. Similarly, the LED stripe in the list is actually longer that the piece needed. Shybo uses a stripe long 30 pixels, while the entire stripe is 144 pixels. So, by dividing the cost for the actual length of the stripe, the cost is reduced from 59,95 $ to 12,5 $. Still regarding the electronics, it is possible to find cheaper alternatives from different brands, for almost each component.

Regarding the design components, instead, the cost in the list is high due to the choice of printing at a professional printing service. Furthermore, these pieces were printed in two steps, each of which required 10 $ of setup costs. Thus, it is possible to presume that by printing in lab the cost of the design components could be significantly reduced, from 53 $ to less than 10 $.

Thus, making the whole prototype in lab and dividing the cost for the actual number of components used, the costs decrease dramatically: from 171 $ to around 80 $. The cost provided in the table, however, provide a complete information for whom, eventually, would like to make Shybo starting from ground without any available materials.

Bill of Materials

Table 2. The table presents the list of the costs and source for each component necessary to build Shybo.

Designator	Component	Number	Cost per unit - currency	Total cost - currency	Source of materials	Material type
Board	Microcontroller board Arduino Pro Mini 328	1	9,95 $	9,95 $	Seeedstudio	Electronics
Potentiometer	Rotatory potentiometer 10K	1	1,28 $	1,28 $	Mouser	Electronics
Button	Colorful round tactile button switch assortment	1	5,95 $	5,95 $	Adafruit	Electronics
Switch	Breadboard friendly SPDT slide switch	2	0,95 $	1,9 $	Adafruit	Electronics
Servo	Micro-servo MG90S High Torque Metal Gear	1	9,95 $	9,95 $	Adafruit	Electronics
LED stripe	NeoPixel Digital RGB LED Strip 144 LED - 1m White - WHITE	1	59,95 $	59,95 $	Adafruit	Electronics
LED ring	NeoPixel Ring 8 x VS2812 5050 RGB LED	1	4,95 $	4,95 $	Trossen Robotics	Electronics
Protoboard	Double-sided Protoboard breadboard 3x7 cm	1	1,09 $	1,09 $	Aliexpress	Electronics
Bluetooth	HC-06 Bluetooth Module	1	2,92 $	2,92 $	Aliexpress	Electronics
Charger	Power Cell - LiPo Charger/Booster	1	19,95 $	19,95 $	Sparkfun	Electronics
Hat	3D printed component	1	11,70 $	11,70 $	Ff3dm.com	PLA
Back shell	3D printed component	1	11,45 $	11,45 $	Ff3dm.com	PLA
Front shell	3D printed component	1	9,44 $	9,44 $	Ff3dm.com	PLA
Base	3D printed component	1	6,75 $	6,75 $	Ff3dm.com	PLA
Axis 1	3D printed component	1	2,91 $	2,91 $	Ff3dm.com	PLA
Axis 2	3D printed component	1	2,95 $	2,95 $	Ff3dm.com	PLA
Servo docking	3D printed component	1	3,88 $	3,88 $	Ff3dm.com	PLA
LED structure	3D printed component	1	4,2 $	4,2 $	Ff3dm.com	PLA

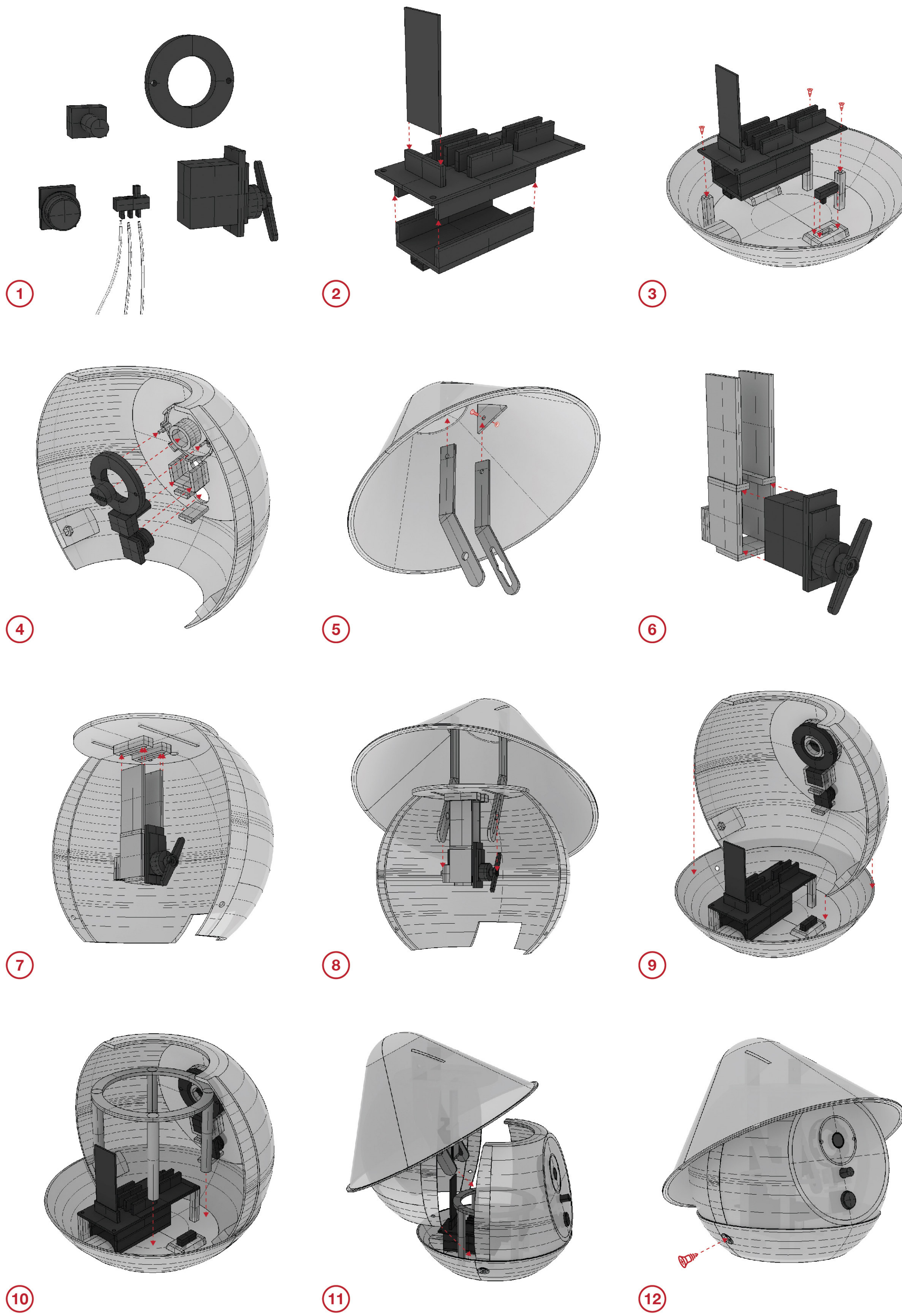
5. Assembly

The assembly of Shybo is not difficult neither risky. However, it may require both time and patience, because of the need for wiring all the components and setting up the small protoboard. Figure 8 shows the twelve main steps of the Shybo's assembly process.

Assembly process:

1. Wire each component using jumper cables, it makes easy to assembly and disassembly every time you need.

2. Setup the protoboard by placing the headers needed for the various components. Then connect the Arduino Pro Mini board on the bottom and the Bluetooth on top. Be aware that the board have to be placed on one side, not in the middle.
3. Connect this group of components to the base that is provided of three small pillars. Use screws to fix the board on the pillars. At the same time place also the switches and connect also their cables to the board.
4. Place the LED ring, the potentiometer and the button on the front shell. The order of the elements is determined by the holes and structure. The first prototype was developed also placing a microphone at the center of the LED ring, but it was not used at this stage. The sound sensing is entrusted to a computer.
5. Connect the axes to the hat with screws and bolts. Axes 1 goes on the right while axes 2 goes on the left.
6. Introduce the servo into the servo docking structure.
7. Place the servo docking structure into the lodging on the back shell.
8. Connect the hat with axes to the back shell with the servo docking. The two axes must slightly flex to be positioned.
9. Locate the front shell on the base and connect the LED ring, the potentiometer and the button cables to the board.
10. Fix with double-sided tape or hot glue the LED stripe to the LED structure. Then place the LED structure on the base, where was previously placed to the front shell.
11. Locate also the back shell on the base. It is necessary to flex slightly the shells to close the robot.
12. On the right and the left of the robot there are two holes. Use two screws to lock the shells and the base.



13.

Fig. 8. Shybo's assembly process.

6. Operation Instructions

Once the assembly of both electronics and the 3D printed parts is complete, and the Arduino sketch is uploaded on the robot, it is possible to use it. Operating Shybo requires three main steps: initialization, train and play.

The *initialization* is a preliminary stage that consist of connecting the robot to a computer via Bluetooth and of initializing both the middleware and the Wekinator.

1. This stage requires the following steps:
2. Open the Processing file called *realTimeSoundSpectrum.pde and run the sketch;*
3. Open the Wekinator software;
4. When Wekinator asks to create a new project, open the *WekinatorProject.wekproj* by going to file and then open project;
5. Connect Shybo to the computer via USB to power it;
6. Connect Shybo to the middleware via Bluetooth serial;
7. OPTIONAL: you can change the minimum sound threshold by moving the slider on the right. Try to set it up so that the ambient sounds are excluded from the recording.

8. **Fig. 9.** From the left: colour selection (A), sound recording (B), and scared status (C).

The *recording* is a stage that allows to associate sounds to the color classes. It is needed at least the first time Shybo is used, since it doesn't have preset color-sounds combinations. New training can be done at any moment by switching status.

8. This stage requires the following steps:
9. Change status through the switch located on the left side of the Shybo's bottom surface*;*
10. Select a colour class by turning the nose, namely the potentiometer (figure 9 A);
11. Record a sound by pressing the mouth, namely the button (figure 9 B). It stops recording when the button is released;
12. Change again the position of the switch to save the trainings;

In the *play* stage Shybo reacts to sounds according to the training. There is only one preset behavior: the scared status (figure 9 C). In this status, activated by sounds that get over a certain threshold (650), the robot closes the hat, blinks in red and shakes. All three stages are easy, fast and safe and no programming skills are required to operate Shybo.

7. Validation and Characterization

As in the case of hardware mentioned in the introduction, Shybo can be used for with three different purposes. On the one hand, it can be employed in research studies, and on the other hand it can support real applications. Regarding the research studies, Shybo can be used to support investigations on robot's features and on human cognitive abilities and perception.

In the first case, the studies could focus on the minimal non-verbal behaviors and on how these are perceived by people. For instance, the robot currently presents two main emotional statuses. Further studies could employ it with specific color-sound combinations aimed at communicating different robot statuses, for evaluating the efficacy of these minimal behaviors. The results of these studies can contribute to the background knowledge to which roboticists might refer for informing their design process. The work by Zaga et al. [2017] represents an example of this kind of study.

In the second case, the robot might be used as a tool to investigate cognitive abilities and perception, as in the study by Boccanfuso et al. [2016] with children affected by Autism Spectrum Disorder. In this kind of studies, the minimal robot behaviors can be used as a way to investigate users' responses to stimuli and approaches to interaction, which can be detectors of cognitive disabilities or learning issues.

Lastly, the robot might be used in real applications for carrying out playful learning activities with children. This prototype, in fact, has already been tested in a school with a group of twelve children. In this kind of applications, the Shybo can be used as part of stories around which investigators, educators, or teachers can build an educational experience. In the first application, in fact, Shybo was used to engage children in a two-hours activity aimed at letting children (6 and 7 years old) familiarize with concepts of color theory and play musical instruments. For this kind of activity, a set of elements, mostly made of printed paper, were built. Thus, Shybo enables playful learning experiences that, every time, can be customized according to the educational interests.

Fig. 10. In the picture a girl is training Shybo as part of a playful learning activity carried out in a primary school.

Finally, Shybo is versatile. Despite its sensitivity to sounds, it is possible to employ in very different situations, that can be characterized by different background noise. For instance, in addition to the application at the school, Shybo was exposed at two public events (e.g. maker fair) in which the environment was very crowded and the level of background noise was very high. In spite of this, it was possible to train it and play with the trained sounds.

Acknowledgments: Shybo was developed within a project supported by Jol CRAB lab, by TIM and with the collaboration of X-Studio, Tsinghua University, China.

References:

[1] J. Fink, S. Lemaignan, P. Dillenbourg, P. Rétornaz, F. Vaussard, A. Berthoud, and K. Franinović. Which robot behavior can motivate children to tidy up their toys?: Design and evaluation of ranger. In *Proceedings of the 2014 ACM/IEEE international conference on Human-robot interaction*, 2014, 439-446.

[2] C. Zaga, R. A. de Vries, J. Li, K. P. Truong, and V. Evers. A Simple Nod of the Head: The Effect of Minimal Robot Movements on Children's Perception of a Low-Anthropomorphic Robot. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, 2017, 336-341. ACM.

[3] S. Yohanan, and K. E. MacLean. Design and assessment of the haptic creature's affect display. In *Proceedings of the 6th international conference on Human-robot interaction*, 2011, 473-480.

[4] F. Michaud, J. F. Laplante, H. Larouche, A. Duquette, S. Caron, D. Létourneau, and P. Masson. Autonomous spherical mobile robot for child-development studies. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *35*(4), (2005), 471-480.

[5] L. Boccanfuso, E. Barney, C. Foster, Y. A. Ahn, K. Chawarska, B. Scassellati, and F. Shic, (2016, March). Emotional robot to examine different play patterns and affective responses of children with and without ASD. In Proceedings of the *11th ACM/IEEE International Conference on Human-Robot Interaction (HRI),* 2016, 19-26.

[6] H. Kozima, M. P. Michalowski, and C. Nakagawa. Keepon. *International Journal of Social Robotics*, *1*(1), (2009), 3-18.

[7] S. Kurkovsky. Mobile computing and robotics in one course: why not?. In *Proceedings of the 18th ACM conference on Innovation and technology in computer science education*, 2013, 64-69.

[8] S. Hussain, J. Lindh, and G. Shukur, The effect of LEGO training on pupils' school performance in mathematics, problem solving ability and attitude: Swedish data. *Journal of Educational Technology & Society*, *9*(3) (2006).

[9] C. R., Ribeiro, C. P. Coutinho, and M. F. Costa. Robotics in child storytelling. In Proceedings of the 6th International Conference on Hands-On Science for All, HSCI, 2009, 198-205.

[10] F. B. V. Benitti, and N. Spolaôr,. How Have Robots Supported STEM Teaching?. In *Robotics in STEM Education,* (2017), 103-129.

[11] A. Bulgarelli, G. Toscana, L. O. Russo, G. A. Farulla, M. Indaco, and B. Bona. A low-cost open source 3D-printable dexterous anthropomorphic robotic hand with a parallel spherical joint wrist for sign languages reproduction. *International Journal of Advanced Robotic Systems*, *13*(3), (2016) 1-12.

[12] S. Woods. Exploring the design space of robots: Children's perspectives. *Interacting with Computers*, *18*(6), (2006) 1390-1418.

[13] P. Baxter, J. Kennedy, E. Senft, S. Lemaignan, and T. Belpaeme. From characterising three years of

HRI to methodology and reporting recommendations. In *The Eleventh ACM/IEEE International Conference on Human Robot Interaction*, 2016, pp. 391-398.

[14] J. Gonzalez-Gomez, A. Valero-Gomez, A. Prieto-Moreno, and M. Abderrahim, A new open source 3d-printable mobile robotic platform for education. *Advances in autonomous mini robots*, (2012) 49-62.

[15] D. E. Atkins, J. S. Brown, and A. L. Hammond. *A review of the open educational resources (OER) movement: Achievements, challenges, and new opportunities,* (2007), 1-84.

[16] M. L. Lupetti, (2017, August 3) SHYBO. Retrieved from osf.io/xk2r4 DOI 10.17605/OSF.IO/XK2R4

[17] M. L. Lupetti, Y. Yao, J. Gao, H. Mi, and C. Germak. Design for Learning Through Play. An Exploratory Study on Chinese Perspective. In *International Conference on Cross-Cultural Design*, 2017, 565-581.

[18] M. L. Lupetti, J. Gao, Y. Yao, and H. Mi. A scenario-driven design method for Chinese children edutainment. In *Proceedings of the Fifth International Symposium of Chinese CHI*, 2017, 22-29.

[19] L. Romagnoli, lorenzoromagnoli/fft_Arduino_wekinator: release [Data set]. http://doi.org/10.5281/zenodo.580300 (2017). prototype Zenodo.

[20] R. Fiebrink, D. Trueman, and P. R. Cook. A Meta-Instrument for Interactive, On-the-Fly Machine Learning. In *NIME* (2009), 280-285.