Digital Fabrication and Maker Movement in Education

eCraft2Learn

Making Computer – supported Artefacts from Scratch

Deliverable D4.2

Tools for a digital maker space in schools

This project has received funding from the European Union’s Horizon 2020 Research and Innovation Programme under Grant Agreement No 731345.

PROJECT DESCRIPTION

Acronym:	**eCraft2Learn**
Title:	Digital Fabrication and Maker Movement in Education: Making Computer-supported Artefacts from Scratch
Coordinator:	University of Eastern Finland
Reference:	731345
Type:	RIA
Program:	HORIZON 2020
Theme:	Technologies for Learning and Skills
Start:	01. January, 2017
Duration:	24 months
Website:	http://www.project.ecraft2learn.eu/
E-Mail:	office@ecraft2learn.eu
Consortium:	**University of Eastern Finland**, Finland, (UEF), Coordinator
	Edumotiva, Greece (EDUMOTIVA)
	Mälardalen University of Sweden, Sweden (MDH)
	Zentrum für Soziale Innovation, Austria, (ZSI)
	The University of Oxford, United Kingdom, (UOXF)
	Synyo GmbH, Austria, (SYNYO)
	University of Dundee, Scotland, (UNIVDUN)
	University of Padua, Italy, (UNIPD)
	Technopolis City of Athens, Greece (TECHNOPOLIS)
	Evothings, Sweden (EVOTHINGS)
	Arduino, Sweden (ARD)
	Ultimaker, United Kingdom (ULTIMAKER)

DELIVERABLE DESCRIPTION

Number:	**D4.2**
Title:	**TOOLS FOR A DIGITAL MAKER SPACE IN SCHOOLS**
Lead beneficiary:	**UNIPD**
Work package:	WP4
Dissemination level:	Confidential (CO)
Type	Report (R)
Due date:	30.06.2017
Submission date:	[30.06.2017]
Authors:	**Emanuele Menegatti, Michele Moro,** UNIPD
Contributors:	**Hanna Nygren**, UEF, **Calkin Suero Montero**, UEF, **Marie Ehrndal,** ARD, **Dimitris Loukatos**, Edumotiva, **Sonya Horton**, ULTIMAKER, **Andrea Alessandrini,** UNIVDUN, **Baran Cürüklü**, MDH**, Afshin Ameri**, MDH, **Ken Kahn**, UOXF, **Christian Voigt**, ZSI, **Margit Hofer**, ZSI, UNIPD, **Francesca Agatolio**
Reviewers:	**Ilkka Jormanainen,** UEF

Version Control

Version	Date	Person in charge (Organization)	Changes	Quality Assurance
1	20.06.2017	Emanuele Menegatti [UNIPD]	created layout and requested inputs	UEF
2	26.06.2017	Emanuele Menegatti [UNIPD]	checked added text	UEF
3	28.06.2017	Emanuele Menegatti [UNIPD]	homogenize document	UEF
4	29.06.2017	Ilkka Jormanainen & Calkin Montero [UEF]	revised and QA	UEF
5	30.06.2017	Emanuele Menegatti [UNIPD]	finalized document	UEF

Acknowledgement: This project has received funding from the European Union’s Horizon 2020 Research and Innovation Action under Grant Agreement No 731345.

Disclaimer: The content of this publication is the sole responsibility of the authors, and does not in any way represent the view of the European Commission or its services.

TABLE OF CONTENT

TABLE OF FIGURES

TABLE OF TABLES

EXECUTIVE SUMMARY

This deliverable reports the first implementation of the hardware and software core technologies selected to implement the eCraft2Learn ecosystem. The main aim of this document is freeze a first implementation of the technical core of the ecosystem, which will be used in the first pilot activities. The pilot activities will provide feedback on this first implementation and a revision will follow. The currently selected software and hardware devices have been integrated, tested and demonstrated. This report describes in detail the first ecosystem, how it can be installed and provides a link to a video with a demonstration of the softwares running on the selected development platforms (namely, Raspberry PI 3).

1 INTRODUCTION

The last years witnessed an increased development and spreading of technologies that radically changed the way people live and work. Scholastic institutions in Europe have been, albeit slowly, partly involved in this phenomenon, not least in the interest of institutions that have support this through specific financial support and supporting research projects on these topics. The recent literature highlights a wide set of projects aimed at documenting and evaluating the impact of the ICT (Information and communications technologies) employment in education [1].

However, despite the abovementioned positive effects, the spreading of ICT in school environments and the increasing pressure of the institutions, the educational system is not yet able to efficiently integrate the ICTs into curriculum paths to improve the cognitive skills of the students. This fact has been exemplified in a report of the European Commission (2013), which stated that "the challenge is not the lack of technology in schools but the huge variation of using technology meaningfully in education"[2]. In fact, new technologies are simply used to reinforce the old ways of teaching – an ineffective approach in today's society, where school kids will be likely called to take jobs that have not been invented yet [3, 4]. This consideration takes an important meaning, especially given the youth unemployment challenge that Europe is facing.

In this document, we will detail the hardware and software setup proposed by the consortium of the eCraft2Learn project to effectively foster the learning of the new skills of the 21st century: digital design in 3D, 3D printing and programming computers and electronic circuits.

In the following, the first technical core of the eCraft2Learn ecosystem will be illustrated. In particular, this documents reports on the testing and evaluation of the hardware equipment, such as 3D printers Arduino and Raspberry Pi boards.

2 TOOLS

In Deliverable D4.1 (Architecture analysis and specification) we provided a broad overview of technologies which are state-of-the-art in 3D Printing, DIY electronics and related software environments. In this document, we report on the selected hardware and software tools creating the technical core of the eCraft2Learn ecosystem. The eCraft2Learn technical core will be extended in the next months with more advanced tools for more advanced users (or classes) if required. These more advanced tools could cover specific subjects of interest of a specific discipline or schools or technology the teachers want to investigate. The current core of tools is supporting the co-design and co-creation of computer supported artefacts for transversal activities and most subjects.

The selection of hardware and softwares composing the technical core of the eCraft2Learn ecosystem has been performed with a thorough analysis of the available alternatives (discussed in Deliverable D4.1) by comparing them with respect to the main goals of **deployment** (in the real schools) and **acceptability** by the schools and/or the users to create a "*sustainable ecosystem*", as well as **supporting the pedagogical stages** of the eCraft2Learn ecosystem. During this analysis the major constraints we considered are:

- Most of the schools have limited financial resources available to buy hardware and software tools. In some countries (e.g., Italy and Greece) schools have almost no budget for ICT. Most of the time the schools rely on external providers. Thus, the target solution should require a small budget.
- Very rarely technical skills for setting up, installing and maintain specific hardware and software tools are available within the school. Thus, the target solution needs to be robust and provide a simple way to recover from failure or file damages.
- There is huge variability of hardware setup of the computer laboratory in different schools. UNIPD has extensive experience collaborating with schools in classes with the latests iMac desktop computers as well as desktop PC running Windows XP. Thus, the target solution should be independent from the operating system and the hardware already available at schools.
- The student (or his/her family) should replicate its own personal hardware and software tools as used in the eCraft2Learn ecosystem and continue to explore ICT also at home and in informal learning settings using the same hardware and software. Thus, the target solution should have a very low cost (i.e. affordable also by each single student) and should be widely available.
- It is of paramount importance that the adopted solution has a large online user community, which can provide active online support and offer use cases, fixes to known issues and, very important for the students' motivation, an audience for sharing the hardware and software artifacts. Thus, the target solution should be widely adopted by the maker community and other schools.

The final choice of the hardware and software configuration is presented and justified in the remainder of this document and it is depicted in Fig. 1. Basically, each student or group of students will work on a developing platform composed of a Raspberry Pi 3 connected to one or more Arduino boards (forming a work station) and they will share a common 3D printer. The single elements will be described in the next sections.

2.1. HARDWARE

The initial experimentations of these first six months has been devoted to prepare and demonstrate the effectiveness of the setup proposed in the followings. The proposed setup is the result of the careful analysis made in D4.1 and of the tests performed for the demonstration of D4.2 (this document). This first integration effort already brought us to revise some initial solutions and to change our mind with respect to some strategic choices. We decided to freeze this first implementation of the technical core of the eCraft2Learn ecosystem as made of:

- a Raspberry Pi 3 (RPi3) as the experimental development platform,
- one or more Arduino boards connected with a variety of electronic components, which constitute the designed artefact,
- a basic set of software tools integrated with the other parts of the eCraft2learn technical core (i.e., UUI, educational extensions, didactical analytics, etc., see D4.1 section 4). This basic set of software is extendible with more tools dedicated to more advanced users
- a 3D printer used for preparing customised physical components integrated in the creation-programming-sharing workflow. This integration represents another innovative aspect of the eCraft2Learn action

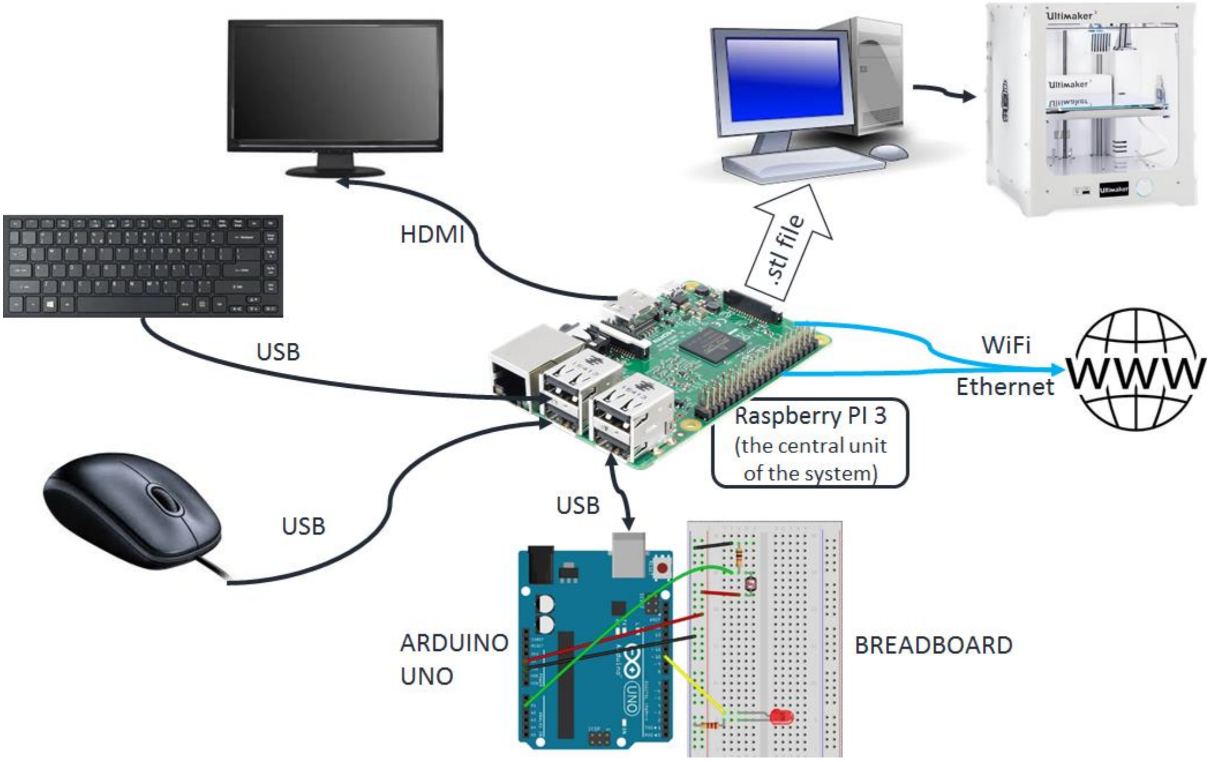


Figure 1. Representative hardware architecture of the technical platform in the eCraft2Learn ecosystem with a PC controlling the 3D Printer for generality. In the next months an implementation with a RPI3 controlling the 3D Printer will be investigated.

Arduino plays a central role as an open-source, low-cost, flexible platform, able to promote the realization of relatively powerful 'intelligent' devices (see D4.1, section 3.5). Though we still are open to other solutions with equivalent or greater positive qualities in terms of availability, easiness of development, affordable cost and flexibility, Arduino currently remains the most convenient solution for providing such platform. The large user community developing software environments for Arduino and its ecosystem (see D4.1, section 3.3) reinforces this choice.

One innovative aspect in the way that we propose to use the Arduino board resides on the its integration with a **Raspberry Pi 3 (RPi3) board** (see D4.1 section 3.5), as the experimental development platform instead of a standard PC. We validated the choice of selecting RPi3 also with respect to the following implementation issues:

- A RPi3 has sufficient computational power and resources to run a Linux distribution: therefore, connecting a mouse, a keyboard and an HDMI monitor to the board renders it a low-cost PC-equivalent, though with some limitations;

- We observed that often an educational software is initially provided for the most common proprietary OS (Windows and MacOs), but soon a Linux version is then developed to favor those school settings that particularly sensitive to the cost factor. After this usually these softwares are ported also to RPI architecture.

- Web-based tools are of particular interest due to their potential to be executed 'anywhere' without the necessity to install or update specific software on the development platform (this is true in general terms, although sometimes specific browser plug-ins are required).

Resource limitations on the RPi3 board are being carefully evaluated in order to provide a real feasible solution. Moreover the proposed architecture, depicted in Fig. 1, will be easily portable on future backward compatible hardware provided some general features are maintained (in terms of variety of connections, sufficient computational power and memory, etc).

The **3D printer** will be used to create customised components when recycled parts or low cost components are not available. The final solution will be to control the 3D printing via a RPI3 . However, running the slicer software[1] on RPI presents some constraints and limitations, as a full operative version of this type of software on RPi3 is not yet available. This is evolution is currently under testing by the partners Ultimaker and UNIPD. However, considering that in a school class there will not be a single 3D printer for each development station (i.e. each RPI3), during the first pilot activities the station controlling the 3D printer will be a standard PC with suitable software drivers. This compound will be seen by users as a shared 3D printing service, which could be also used as stand alone facility for other projects in the school. However, the UUI will guarantee a easy working environment for the practitioners in spite of the different technologies and approaches used. We will work towards enabling a RPi3 unit to function as a controlling station, since this would represent in the long term a lower cost solution to the schools when deploying the eCreaft2Learn technical platform.

2.1.1. Raspberry Pi (RPi)

Raspberry Pi [https://www.raspberrypi.org/] is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python. It's capable of doing everything you'd expect a desktop computer to do, from browsing the internet and playing high-definition video, to making spreadsheets, word-processing, and playing games. Furthermore, the Raspberry Pi has the ability to interact with the outside world,

[1] software used for printing designed objects with a 3D printer

and has been used in a wide array of digital maker projects, from music machines and parent detectors to weather stations and tweeting birdhouses with infra-red cameras. The wide adoption of RPI3 and the enormous amount of examples available will inspire the student in the **ideation** stage and will engage them in **sharing** their artefacts with the rest of the world.

We selected Raspberry Pi 3 (RPi3) (fig. 2) for this project, because about 50% faster than the Raspberry Pi 2. RPI3 is mounting a BCM2837 chip. The underlying architecture of the BCM2837 is identical to the BCM2836. The only significant difference is the replacement of the ARMv7 quad core cluster with a quad-core ARM Cortex A53 (ARMv8) cluster. The ARM cores run at 1.2GHz. The Videocore IV runs at 400Mhz. Regarding the power needs the Raspberry Pi 3 is powered by a +5.1V micro USB supply. Exactly how much current (mA) the Raspberry Pi requires is dependent on what one connects to it, but normally a standard 2.5A power supply will provide ample power to run Raspberry Pi. The power requirements of the Raspberry Pi increase as the various interfaces on the Raspberry Pi are used. The GPIO pins can draw 50mA safely, distributed across all the pins; an individual GPIO pin can only safely draw 16mA. The HDMI port uses 50mA, the camera module requires 250mA, and keyboards and mice can take as little as 100mA or over 1000mA. One powerful feature of the Raspberry Pi is the row of GPIO (general purpose input/output) pins along the top edge of the board. The pins can be programmed to enable the interaction of the artefacts with the real world. Inputs don't have to come from a physical switch; it could be input from a sensor or a signal from another computer or device, for example. The output can also perform different operations, from turning on an LED to sending a signal or data to another device. If the Raspberry Pi is on a network, devices that are attached to it from anywhere can be controlled and those devices can send data back. Connectivity and control of physical devices over the internet is a powerful and exciting thing, and the Raspberry Pi is ideal for this. The large data set of examples developed by the user community of RPI3 will foster creativity in the eCraft2Learn students.

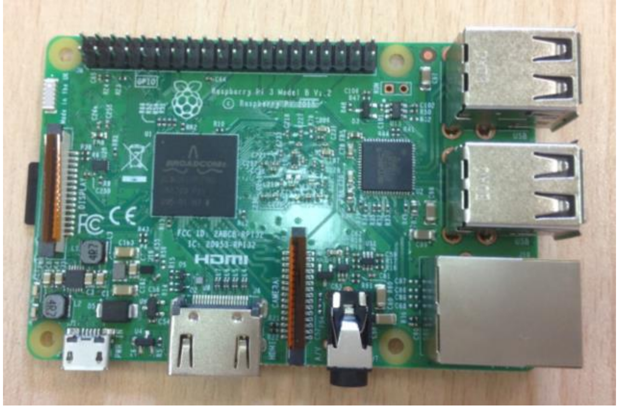


Figure 2. Raspberry PI 3

The abovementioned characteristics of the Raspberry Pi board, its wide acceptance and its active support community are leading us to select this specific board to play a dominant role in the project.

More specifically, one of the main objectives of the eCraft2Learn project is the young students participating in the eCraft2Learn pilots to use Raspberry Pi units [https://www.raspberrypi.org/products/raspberry-pi-3-model-b/] running raspbian Jessie [https://www.raspberrypi.org/downloads/raspbian/] instead of conventional PCs. These Raspberry Pi units should be accompanied by a keyboard/mouse set and a TFT screen. **The adoption of this solution, amongst the other benefits is that it tackles the problems related with the vast software and hardware diversity characterizing a typical School Lab and reduces the Lab's overall energy**

consumption. Ethernet (IEEE 802.3) or WiFi (IEEE 802.11) connections on the Raspberry Pi units will be active in order to provide Internet access to each student workstation and to support remote communication (file sharing, remote 3D printer control, communication with the artefacts being made).

We are targeting to develop the technical core of the eCraft2Learn ecosystem entirely based on Raspberry Pi 3, hence we would like to use also an additional Raspberry Pi 3 unit to control the task of 3D printing slicing the 3D model and sending commands to the 3D printer, instead of using a standard PC dedicated to this job. However, as explained earlier, difficulties arise since a universal slicer software for every 3D printing is not yet available for RPi3. For some printers, like Ultimaker 2 , a possible solution is based on Octopi software (a modified version of Raspbian Jessie lite supporting Octoprint and the Cura slicing engine) [https://octopi.octoprint.org/], further details are given in section 3D Printer Ultimaker. The Raspberry Pi unit supporting the printer should play the role of the Internet gateway and the file server for the rest of the Lab's island or can host a web server as well. A Raspberry Pi unit equipped with the Raspbian operating system supports the Apache web server as well as various Python based HTTP servers that can be very useful in communication with other modules. Unfortunately, not all 3D printer are compatible with Octopi. For instance Ultimaker 3 is not and this calls for a specific software development. As written before, we will work in the next months to overcome this problem.

Each Raspberry Pi unit is equipped with the MIT Scratch software and the Scratch GPIO module [https://www.raspberrypi.org/documentation/usage/scratch/gpio/README.md] to provide direct access to the GPIO pins of the RPI. Python GPIO libraries like the gpiozero [https://gpiozero.readthedocs.io/en/stable/] can be useful as well. For 3D printing task (creation of STL files) the FreeCAD tool [https://www.freecadweb.org/] should be used. This tools runs well on the RPIs.

Additionally, on each Raspberry Pi unit the Arduino IDE [https://www.arduino.cc/en/Main/Software] and the Ardublock tool [http://blog.ardublock.com/] are installed. This fact allows users to program one or more simple Arduino units [https://www.arduino.cc/en/main/arduinoBoardUno] via the USB interface of the Raspberry Pi; each Arduino is perceived by the Raspberry Pi as a /dev/ttyACMn device, where n is a physical number starting from 0 for the 1st arduino plugged in the Raspberry Pi. Snap4Arduino should work provided that the StandardFirmdata firmware Arduino sketch is installed. Furthermore, several web-based and/or visual block programming tools will work on the Raspberry Pi units.

Artefacts created by students can be programmed/controlled remotely by the workstation Raspberry PIs (via Bluetooth or WiFi). In addition to this, the presence of one or more android tablets/phones (with gyro, magnetic, vibration sensors) should provide a good and more flexible way to control robots and / or interact with Arduino / Raspberry Pi units via Bluetooth or WiFi interfaces, as well. The well known MIT AppInventor [http://appinventor.mit.edu/explore/] web based visual block programmer for Android devices (or the Pocket Code [https://play.google.com/store/apps/details?id=org.catrobat.catroid&hl=en]) can be used to program these tablets/phones. The artifact units (Raspberry Pi assisted or not) may be run on power banks with solar assistance for better autonomy and environmental reasons. Raspberry Pi based autonomous artefacts are also capable of performing complex scenarios. Cheap sensors and actuators for Arduino, Raspberry Pi and little breadboards will be present as well, including photoresistors, microphones with preamplifiers, potentiometers, distance sensors, temperature sensors, angle servos, motors, buzzers, leds, L293D drive chips, etc.

2.1.2. Arduino

The eCraft2Learn Raspberry Pi device, that is used as a desktop computer, will be used to communicate with external devices. In most cases, these external devices will be Arduino UNO boards which are exploited to implement computer supported artefacts. The user can program, monitor and communicate with the Arduino UNO boards of these artefacts directly from the RP3 device.

The Arduino hardware used in proposed version 1 of the platform will be the popular Arduino UNO (fig.3) board. The UNO has both digital and analog pins that allow users to control actuators such as LEDs, Piezo components, servo motors, DC motors, and the PWM technique provides some pins with the possibility to fade LED’s, custom control DC motors, vibrators, etc. Input components used can be everything from simple digital buttons, to temperature sensors, light sensors, proximity sensors, motion sensors, variable resistors (such as potentiometers, flex sensors, pressure sensors, etc), to more sensors that are tied to other communication protocols.

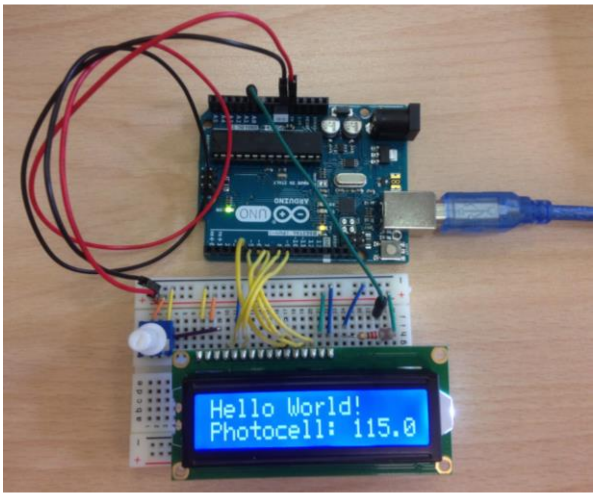


Figure 3. Arduino UNO board connected to a breadboard

The ADC dependent analog input pins and PWM output pins allow for complex sensor and actuator behaviour as interactivity is added to projects. Apart from it’s computing capabilities, the Raspberry Pi has IO pins that can be used to read and control ‘On’ and ‘Off’ states, just like an Arduino board. The Arduino also adds possibilities with it’s features for simple analog input and output, enabling components and devices connected to make use of a span of values. This allows for great complexity in terms of how sensors and actuators can be combined and used as learners experiment and innovate. In addition to this, Arduino can help support independent learning through rich documentation and a widespread community.

The block based code environment envisioned in version 1, that allows learners to shape the behaviour of the Arduino, will be available through the web based UUI. For version 2, an addition of mobile functionality and textual coding is to be explored. This solution would likely involve the WiFi enabled MKR1000 board and an app that allows learners to program using JavaScript. There will also be possibilities to explore the creation and manufacturing of hardware shields of components that can demonstrate relevant electronics concepts, such as analog and digital, input and output, and ‘on’ and ‘off’ states. Other uses for a wirelessly communication enabled device like the MKR1000 or MKRFOX1200 board would be to enable more straightforward IoT innovation.

2.1.3. 3D PRINTER

The making of the artifacts will be facilitated with 3D printing technology. Specifically the Ultimaker 2+ and Ultimaker 3 3D printers (table 1).

Ultimaker 2+

This is the most popular choice of Ultimaker 3D printer currently for schools as it is a great value all-around professional desktop 3D printer that delivers consistent results. Engineered to perform, the Ultimaker 2+ is reliable, efficient, and user-friendly. With a simple display and control dial, even very young students can operate this printer independently with a little prior instruction. The Ultimaker 2 Extended+ offers the additional benefit of an additional 10cm of build height, perfect for larger prints that students may require.

Ultimaker 3

The Ultimaker 3 (fig. 4) is the most reliable dual extrusion 3D printer available, allowing students to print models in 2 colours or materials without pausing the print. This professional desktop printer allows for prints with more complex geometries to be created by using dissolvable PVA for printing the support structures. This provides greater opportunities for student to realise more complex designs.

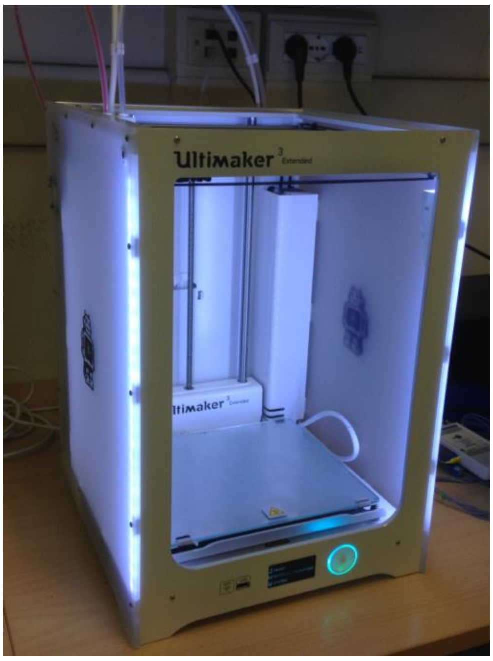


Figure 4 Ultimaker 3 printer

Table 1 Ultimaker 3D printers specifications

Model	**Ultimaker 2+**	**Ultimaker 2+ Extended**	**Ultimaker 3**	**Ultimaker 3 Extended**

Build Volume (max single print size)	223 x 223 x 205 mm	223 x 223 x 305 mm	215 x 215 x 200 mm Dual Extrusion 197 x 215 x 200 mm	215 x 215 x 300 mm Dual Extrusion 197 x 215 x 300 mm
Layer Resolution	From: 600 micron Up to: 20 micron	From: 600 micron Up to: 20 micron	From: 200 micron Up to: 20 micron	From: 200 micron Up to: 20 micron
Build Speed	Up to 24 mm^3/s	Up to 24 mm^3/s	Up to 16 mm^3/s	Up to 16 mm^3/s
Travel Speed	Up to 300 mm/s	Up to 300 mm/s	Up to 300 mm/s	Up to 300 mm/s
Filaments	Open filament system Optimized for: PLA, ABS, CPE, CPE+, PC, Nylon, TPU 95A 2.85mm diameter filaments	Open filament system Optimized for: PLA, ABS, CPE, CPE+, PC, Nylon, TPU 95A 2.85mm diameter filaments	Open filament system Optimized for: Nylon, PLA, ABS, CPE and PVA	Open filament system Optimized for: Nylon, PLA, ABS, CPE and PVA

PLA is the recommended filament for use in schools as it requires no ventilation, is biodegradable, low cost and produces consistently good printing results.

2.2 Software

In D4.1 section 3.3 we provides a first analysis of software tools for programming Arduino-based systems. In addition of the ‘native’ Arduino SDK (IDE, compiler and library) most softwares seem to privilege a block-based orientation, considered more suitable for rapid prototyping and in terms of usability. Our concern here is to identify and integrate a set of tools showing these relevant characteristics:

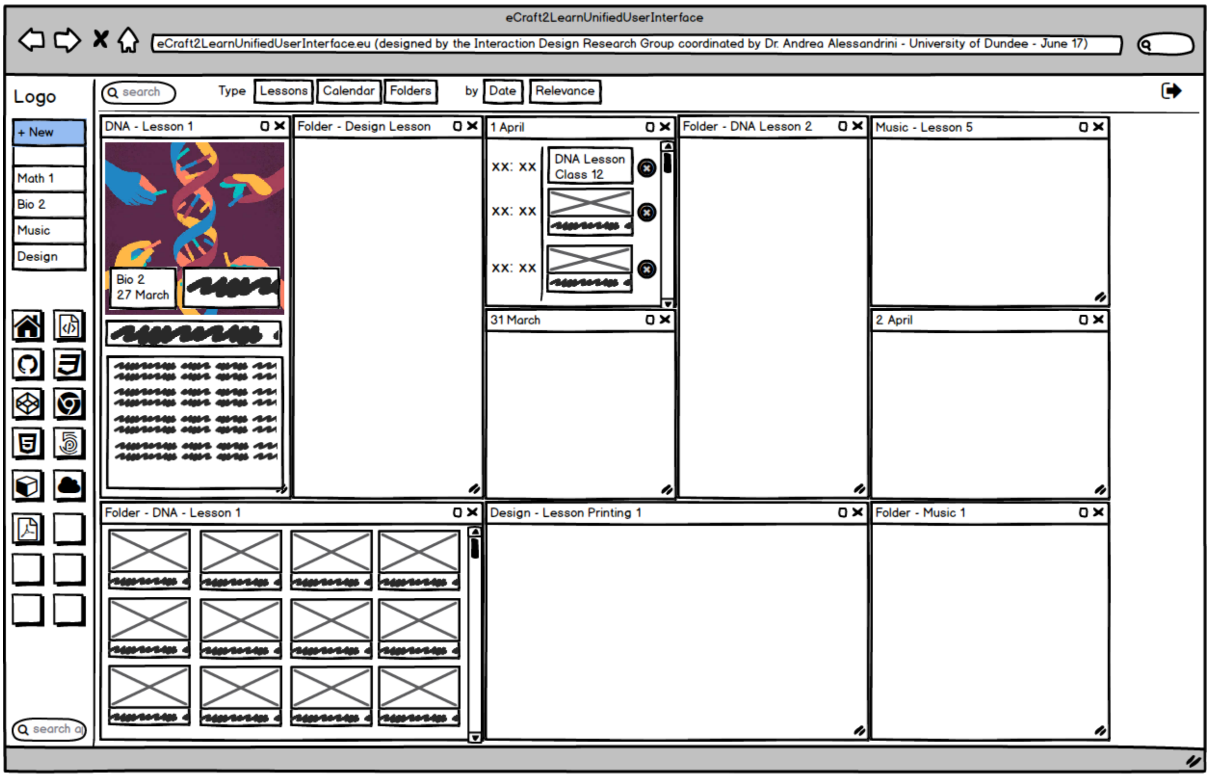
1. runnable on a RPI3 environment;
2. reduced need of installation/update of software elements;
3. friendly user interface;
4. pedagogically meaningful;
5. easiness of integration with the external hardware;
6. open source, free or at least low cost.

Point 2. votes in favor of web-based tools which, on the contrary, may present some limitation in the connection with external hardware and shortcuts must be provided for this purpose in order not to compromise the overall security (though we are dealing with small hardware, RPI3 is fully connectable to the network and security issues should not be neglected). In the following section 3 the proposed set of tools is described together with some details about their installation on a RPI3 system.

As explained in section 2.1, 3D printing services are provided by a shared printing station equipped with the necessary slicer software which completes the features of the printing hardware. On the RPI3 system we need only 3D modeling tools that support the individual design of 3D objects and store the produced models on files with a standard format for successive elaborations. The last printing phase can be supported by a specific feature of the unified user interface to ease the transmission of the 3D model and the activation of the printing service.

To design the user interface, we begun from the initial generic user interface envisioned for the Task

4.1 Architecture analysis and design and described on the deliverable 4.1.



eCraft2LearnUnifiedUserInterface.eu (designed by the Interaction Design Research Group coordinated by Dr. Andrea Alessandrini - University of Dundee - June 17)

Figure. 5 Sketch of the eCraft2Learn User Interface (UI)

The web-based Unified User Interface (UUI) is designed as a dashboard where users might easily add or remove components. We imagined the UUI features a central section where learning contents are available to learners. The interface distinguishes between the coach/teacher with students interactions with the system. The dashboard (fig.5) contains folders, lessons, and events created by teachers. These modules can be resized, moved and, closed. Clicking on lessons switches the view from the dashboard to the “desktop”, which is normally used to operate apps (docs, pdf, etc). Double-Clicking on folders opens the folder in the desktop view for easier navigation. Clicking on a file from the dashboard opens that file in the desktop view. Entries can be removed from the dashboard by clicking on the [x] next to them. We also imagined that the students can access default management and planning tools. Clicking on a calendar entry opens the calendar app at the selected entry in the desktop view. The calendar module in the dashboard can be resized to show more entries at once. On the side of the UUI we had our toolkit section, where students and coaches can choose between tools such Arduino Create, Snap!, and many others. This initial interface design draft will be further evaluated with some users to understand the usability and user experience.

Behind the UUI there is also a user analytics engine to assess and monitor the learning process. The purpose of learning analytics in the eCraft2Learn ecosystem is to provide automated tools to assist learners and instructors in the five-steps process. These analytics tools use well-known educational data mining (EDM) techniques to provide both retrospective and real-time analysis to guide the learning and instructing process. eCraft2Learn ecosystem and different parts of it (hardware and software solutions as described in this document) produce data that are analysed in a separate analytics tool (that is, the analytics tool is not part of the Unified User Interface or background service of it). The analytics environment is platform independent and it is implemented as a browser-based service with HTML5 and JavaScript. EDM algorithms in the tool are selected so that they can handle variety of data types and formats. The data mining process is opened and exposed to the users as much as possible, and the users are also given tools to affect to the process, for example by selecting what data items are relevant in a particular process, what kind of time frames are used in sequential analysis, etc. The environment provides suitable EDM tools for each part of the eCraft2Learn process, and visualizations aim to support teachers' pedagogical understanding about the situation as well as possible.

3 Demonstration of hardware and software

3.1. Installing the system

Raspberry Pi 3 runs with Raspbian Jessie (OS), based on ARM architecture. Each RPI unit is equipped with the MIT Scratch Software, the Scratch GPIO module (that provides direct access to the GPIO pins of the RPI) and Python GPIO libraries.

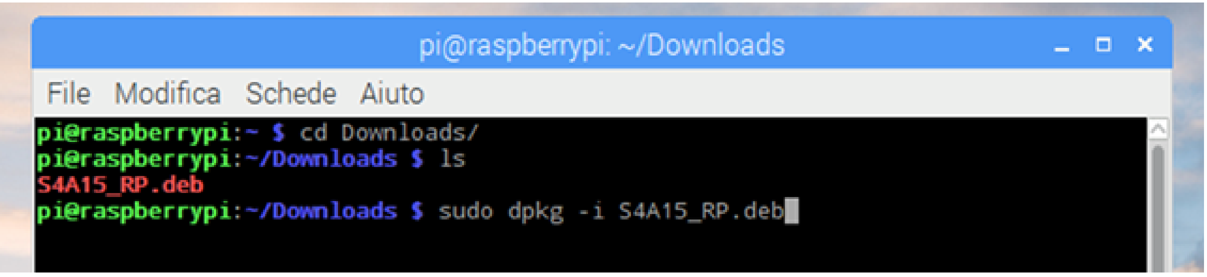
3.1.1. Scratch

Scratch is already installed by default into Raspbian. You can find it in the left-top menu button, in the section "Programmation".

3.1.2. Scratch4Arduino

To install the software you need an internet connection. Steps to install Scratch4Arduino:

- Open the browser (Chromium) and go to www.s4a.cat
- Go to Downloads and select the version "Raspbian (Debian for RaspberryPi)". The file (S4A15_RP.deb) will be saved in the path /home/pi/Downloads
- Open the Terminal
- Type the following commands: "cd Downloads", press Enter and then type "ls", it will appear the downloaded file "S4A15_RP.deb"
- Type "sudo dpkg -i S4A15_RP.deb" and press Enter (fig.6)



```
pi@raspberrypi:~ $ cd Downloads/
pi@raspberrypi:~/Downloads $ ls
S4A15_RP.deb
pi@raspberrypi:~/Downloads $ sudo dpkg -i S4A15_RP.deb
```

Figure 6 Screenshot for the installation of Snap4Arduino

3.1.3. Snap4Arduino

To install the software you need an internet connection. Steps to install Snap4Arduino:

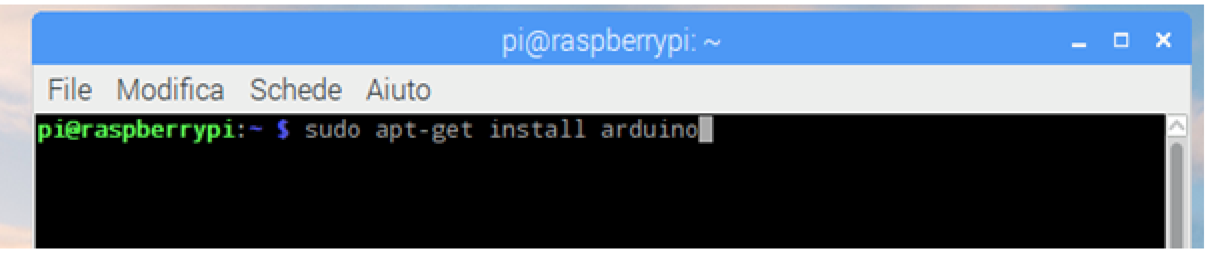
- Go the site http://snap4arduino.org/ and download the ChromeOS version (http://snap4arduino.org/downloads/1.2.3/Snap4Arduino_web-chromeos_1.2.3.zip)
- Extract the file .zip into the "Downloads" folder.
- Open the browser Chromium (next to the menu button)

- In the search bar, type the following url: chrome://extensions.
- Open the extracted folder Snap4Arduino.
- Drag the “snap4arduino.crx” file in the chrome extensions page.
- Accept the install (it may take a few seconds before appear).
- Go to chrome://apps page, and create a shortcut. You will find the Snap4Arduino application into Desktop and menu, in addition to the page chrome://apps.

3.1.4. IDE Arduino

To install the software you need an internet connection. Steps to install IDE Arduino:

- Open the Terminal in the top bar.
- Type the following command: *“sudo apt-get install arduino”* and press Enter (fig.7)



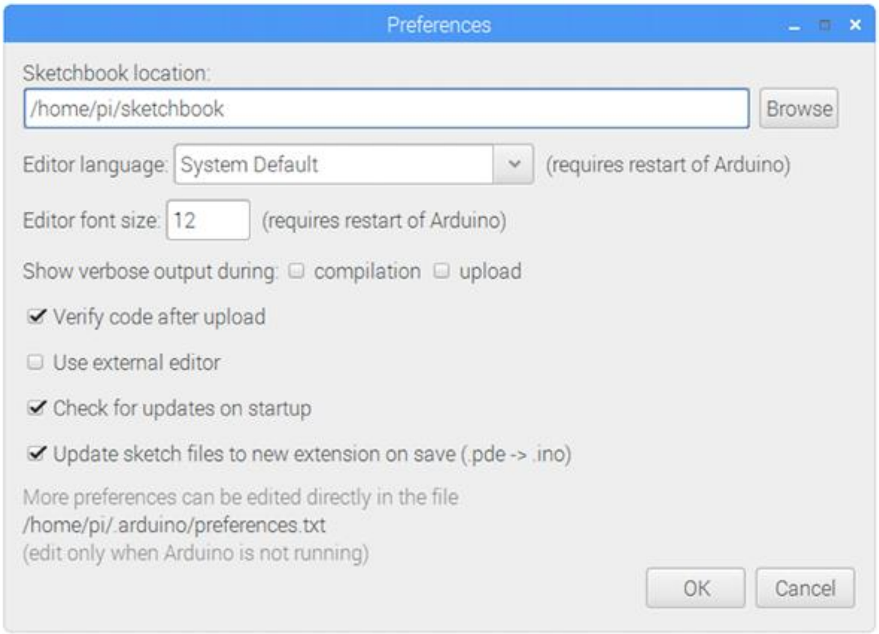
```
pi@raspberrypi:~ $ sudo apt-get install arduino
```

Figure 7 Screenshot for the installation of IDE Arduino

3.1.5. ArduBlock

To install the software you need an internet connection. Steps to install ArduBlock:

- Download “*ardublock-all.jar*” from the following link: http://sourceforge.net/projects/ardublock/files/ardublock-all-20130712.jar/download.
- In Arduino IDE, open menu “Arduino” -> “Preferences” (fig.8)



Sketchbook location: /home/pi/sketchbook

Editor language: System Default (requires restart of Arduino)

Editor font size: 12 (requires restart of Arduino)

Show verbose output during: □ compilation □ upload

☑ Verify code after upload

□ Use external editor

☑ Check for updates on startup

☑ Update sketch files to new extension on save (.pde -> .ino)

More preferences can be edited directly in the file
/home/pi/.arduino/preferences.txt
(edit only when Arduino is not running)

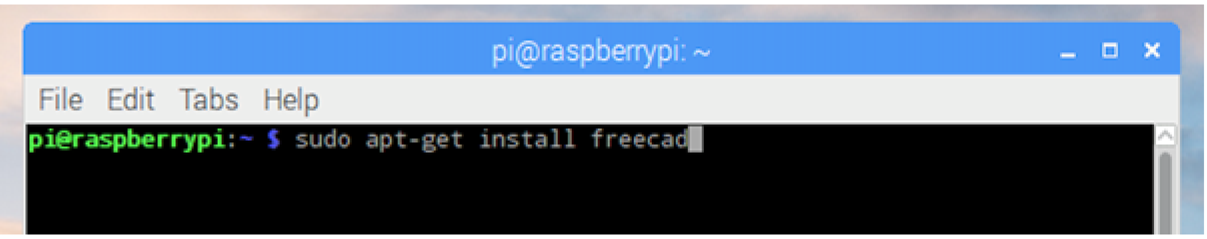
Figure 8 Screenshot for the installation of ArduBlock

- Find “Sketchbook location” (in Raspbian, it’s by default “sketchbook” under user’s home directory).
- Copy “*ardublock-all.jar*” to “*tools/ArduBlockTool/toolardublock-all.jar*” under “Sketchbook location” (In Raspbian /home/pi/sketchbook/tools/ArduBlockTool/tool/ardublock-all.jar).
- Start the Arduino IDE and find ArduBlock under the Tool menu.

3.1.6. FREECAD

To install the software you need an internet connection. Steps to install FreeCAD:

- Open the Terminal in the top bar.
- Once you’ve opened the terminal type the following command: “*sudo apt-get install freecad”* and press Enter (fig.9)



```
pi@raspberrypi:~ $ sudo apt-get install freecad
```

Figure 9. Screenshot for the installation of FreeCAD

3.1.7. CONNECTION WITH ULTIMAKER 3

We tried to connect Raspberry Pi 3 with Ultimaker 3D but this solution seems to be not possible at the moment. Below the steps and the issues that we met.

- Octopi: it is an Open-Source OS, installable on a Raspberry Pi 3, that allows to control some 3D Printers like Ultimaker 1 and 2. It doesn’t work with Ultimaker 3 because Ultimaker 3D printer uses an internal control software, combined to an installable desktop control program called Cura Software (https://ultimaker.com/en/products/cura-software).

- Cura Software: Cura Software is a free program that allows to control Ultimaker printers like Ultimaker 3. It’s available for macOS, Windows and Linux with 32-64bit architecture, but not for Raspberry Pi 3 that has an ARM architecture.

- Wine: to solve the issue about Cura we installed Wine, an Open-Source emulator to run Windows applications on Linux (https://www.winehq.org/download). It runs in a 32bit-64bit architecture but the Raspberry is based on an ARM architecture, so for the installation it needs another program, called Exagear.

- Exagear transforms the Raspberry architecture in a 32bit version and allows to install and run Wine. It is a payment program available for Raspberry Pi Zero, 1, 2 and 3 (https://eltechs.com/product/exagear-desktop/exagear-desktop-features-and-prices/). The installation instructions are available at the following link: https://eltechs.com/run-wine-on-raspberry-pi/.

- Cura Software: we finally install Cura but it appears too slow to be usable on Raspberry PI

We propose to use one **standard pc per class to control the Ultimaker 3D printer**, without using Raspberry Pi 3 during the first round of pilots. We will further investigate in the upcoming months suitable solutions to the encountered problems.

3.2. Testing the System

We tested the system through a simple activity with Arduino using all the software installed (Fig.10) (Arduino IDE, Scratch4Arduino, Snap4Arduino, Ardublock). The activity was to blink a LED connected to a digital PIN of Arduino (fig.6). The activity carried out with all the software. We met some issues with Snap4Arduino due to frequent lags. However using the Berkeley release of Snap! [https://snap.berkeley.edu/snapsource/snap.html] communicating with a local web server can perform at least 20 actions per second.

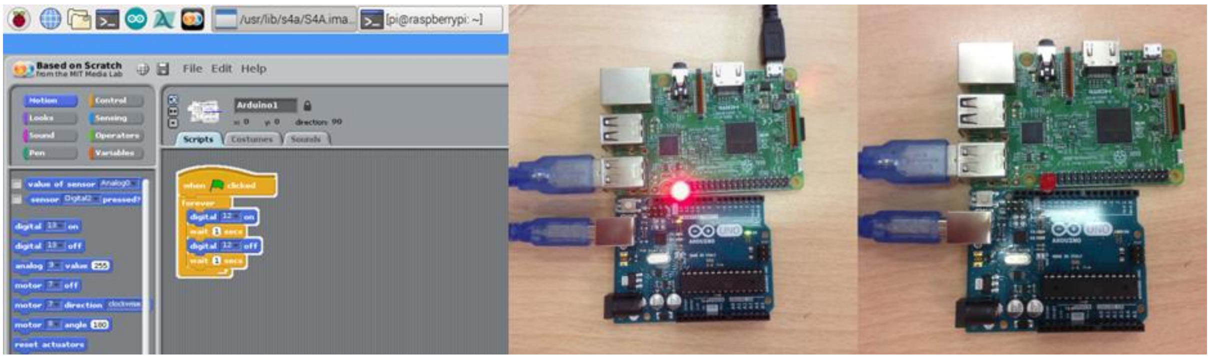


Figure.10. To blink a LED

3.3. Performance analysis

The installed software presents good performances when run on (or through) the RPI3. However, the applications more demanding in terms of graphic rendering (in particular FreeCAD, TinkerCAD and Snap4Arduino) are less responsive and sometimes exhibit a small delay in the interaction with the user. Also, Arduino IDE needs some seconds to open and to download the file on Arduino. In the next months we will investigate if this is just a matter of some settings or if there is a way to improve the smoothness of the user interaction. The performance of web-based applications is very good, even for applications which require graphical rendering (as already mentioned TinkerCAD presents some delays in the user interactions when manipulating reach 3D models).

To have a visual feeling of the performance of the software listed above, we recorded a video of the demonstration of the performances of each software when opened and run from the RPI3. The video can be retrieved at this link:

https://drive.google.com/open?id=0B2mtcmmj3x2TclptaXBDaDlpYjQ

4 PEDAGOGICAL CONSIDERATIONS

The eCraft2Learn pedagogical framework is developed within five stages and is based on the idea of craft- and project-based learning within open learning scenarios. The five pedagogical stages are as shown in Figure 11

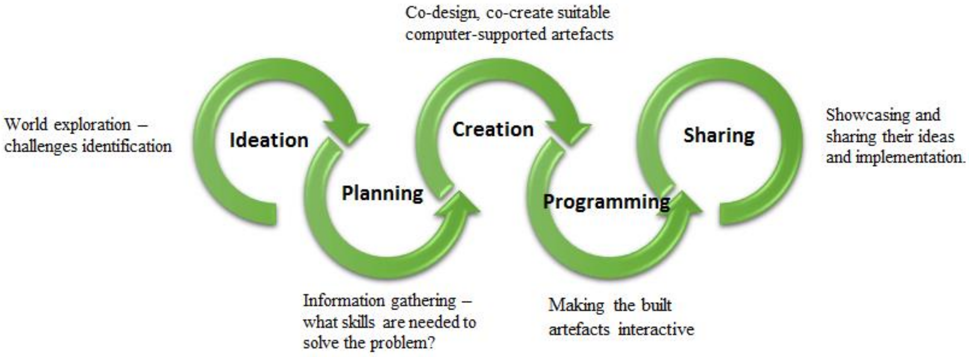

Co-design, co-create suitable computer-supported artefacts
World exploration – challenges identification
Ideation
Creation
Sharing
Showcasing and sharing their ideas and implementation.
Planning
Programming
Information gathering – what skills are needed to solve the problem?
Making the built artefacts interactive

Figure 11. The five pedagogical stages in eCraft2Learn

The students´ learning process is supported in every stage with proper technologies which are a part of the eCraft2Learn ecosystem as well as teachers and experts are. The creation of various types of artifacts is the main focus of the pedagogical stage 3. This stage is vital from the technical perspective described here since it is the first stage during which the learners start the concrete work of making and realising their designs/ideas. This stage is also important from a pedagogical perspective, since it is at the core of the project-based learning, **requiring the integration and use of different skills (e.g., cognitive skills as well as collaboration, communication, teamwork), using the technologies** proposed by the eCraft2Learn action. Thus, one critical pitfall, as identified in D3.1 report, is associated with domain knowledge, i.e. the ability to use the technologies and tools in a satisfactory manner so that they do not become a hindrance in the learning process.

In order to preliminary test the suitable integration of the 5 stage pedagogy with the core technology hardware and software tools we selected for the eCraft2Learn ecosystem, a preliminary eCraft2Learn Workshop was organized in May 2017 in Joensuu SciFest 2017 international event. Five groups of around ten participants (aged 10 to 16) from five different schools were divided in smaller groups to create 'robots heads'. Creating the robots heads proceeded based on the the idea of five stages of craft- and project-based learning methodology (1. Ideation, 2. Planning, 3. Creation, 4. Programming, 5. Sharing). It is to notice that these first three stages seemed to mix during the development of the tasks, and *during this trial these stages were not supported by all the technologies described here (i.e., there was not a 3D printer available)*. Students had all the crafting material on their tables (e.g. paper, carton boxes, toilet paper rolls, scissors, pencils, tape, glue, LEDs, wires, sensors, etc.), as well as a Raspberry Pi computer running Scratch4Arduino (S4A) connected to an Arduino Uno running the S4A firmware, for programming. Students had a chance to explore programming first before starting to define their challenge. Pedagogically, creating and programming were integrated but only one group started with programming. The goal of programming was to make the the eyes of the robot blink. After the eagerness of creating the robot head using all the material on the table, there was a sudden stop before starting to make the programming, so there was a time gap between creation and programming.

The technologies used in the workshops (RPis, Arduino boards, S4A software, and other electronic components) were suitable considering the development and the previous experience of the students. Still, clear and short instructional videos may be needed to help students to perceive what measures has to be done and in which order. This could ease to use the technologies and make it more approachable.

5 Activities examples

Indicative scenario description

The scenarios mentioned below require a Raspberry Pi based workstation for programming and monitoring purposes. Artifacts' power needs are covered by power bank or power supply units. The presence of a 3D printer unit is necessary as well as the configuration of the appropriate 3D printing software on the raspberries.

Table 2 Scenarios description

Scenario #	Scenario Description	Hardware to be used	Software to be used	Students Role
1 "The shy rabbit"	Students are sketching a rabbit on a paper, they put it on an angle servo and move it up or down according to a microphone readings (i.e. a loud sound makes the rabbit to go down).	Arduino Uno, Angle servo, Microphone with amplifier, Some wire cables	Ardublock, Arduino IDE, Snap4Arduino.	To connect microphone and servo on the arduino. To program the arduino using visual blocks. To sketch a rabbit and glue it on the servo.
2 "Sunflower"	Students are 3D printing a flower, they put it on an angle servo and move it around its axis according to photoresistor array readings.	Arduino Uno, Angle servo, 2 or more photoresistor and resistor pairs, Some wire cables	Ardublock, Arduino IDE. Snap4Arduino.	To connect photoresistors and servo on the arduino. To program the arduino using visual blocks. To 3D print a flower and glue it on the servo.

3 “User triggered sound invocation”	a) Students are shaking a tablet, thus invoking a sound to be produced by a pair of speakers connected to an arduino unit. b) Students are repeating the same scenario using a raspberry pi instead of the arduino and the WiFi interface.	a) Arduino Uno, bluetooth, speakers, Android tablet, Some wire cables b) Raspberry pi, (it has WiFi on board), speakers, Android tablet, Some wire cables	a) App Inventor Ardublock, Arduino IDE, b) App Inventor, Simple http server implemented in python	To run .apk on tablets, To program using visual blocks. a) To connect speakers and Bluetooth module on the arduino. b) To connect speakers module on the raspberry. To understand / modify elementary python code
4 “Sensor triggered sound invocation”	Voice message invocation on an rpi unit, according to luminosity sensor readings	a) Raspberry pi, Photoresistor and resistor, Speakers. b) Same as in case a) plus an Arduino UNO used as a sensor board.	a) Scratch and Scratch GPIO b) Snap4Arduino and Arduino IDE	To program using visual blocks. a) To connect photoresistors and speakers to the raspberry board. To load sounds on the Scratch environment. b) To connect photoresistors to the arduino board. To connect speakers and the arduino to the raspberry board. To load the StandardFirmdata firmware using the Arduino IDE. To load sounds on the Snap! environment.

5 “typical robot”	a) Students are 3D printing the chassis of a small robot equip it with to servos and an arduino and drive it through the bluetooth interface of their tablet. b) Students are repeating the same scenario using a raspberry pi instead of the arduino and drive the robot through the WiFi interface of their tablet c) Same as b) but students now drive the robot through the WiFi interface of their workstation using a VNC connection to the remote raspberry (artifact).	a) Arduino Uno, bluetooth, 2 servo motors with wheels, Android tablet, Some wire cables b) Raspberry pi, (it has WiFi on board), 2 servo motors with wheels, Android tablet, Some wire cables c) Same as b) but no tablet is necessary	a) App Inventor Ardublock, Arduino IDE, b) App Inventor, Simple http server implemented in python c) Scratch and Scratch GPIO	To program using visual blocks. To 3D print objects. a) To run .apk on tablets, To connect servos and Bluetooth module on the arduino. b) To connect servos on the raspberry. To understand / to modify elementary python code c) To connect servos on the raspberry.
6 “case for Raspberry”	Students print the cases for the Raspberry units using 3D printer 	Raspberry PI 3 Ultimaker 3D printer	FreeCAD	To print 3D print objects To construct the components for their hardware

6 CONCLUSION

This document presented the demonstration of the first implementation of the technical core of the eCraft2Learn ecosystem. The eCraft2Learn ecosystem is an integrated solution of hardware and software to foster learning of the 21st century technology skills by students in both formal and informal learning. This document presents the hardware devices and the software tools selected among the ones investigated in Deliverable D4.1. The selection was motivated considering the practical issues rising when deploying the ecosystem in the schools and the effectiveness of the pedagogical approach behind each tool. We detailed the integration and the testing of the ecosystem and we demonstrated, even with a video, that the first version of the ecosystem is running and is documented. After the first pilot activities, all the choices made so far will be discussed and validated with the user feedback and if necessary revised. This in accordance with the "design thinking" methodology we adopt in this project.

7 REFERENCES

Literature:

[1] Button, D., Harrington, A., & Belan, I. (2014). E-learning & information communication technology (ICT) in nursing education: A review of the literature. Nurse Education Today, 34(10), 1311-1323.

[2] European Commission, 2013, "Survey of schools: ICT in education"

[3] Fullan, M., & Langworthy, M. (2013). Towards a new end: New pedagogies for deep learning. Seattle, Washington: Collaborative Impact.

[4] Gülbahar, Y., & Kalelioğlu, F. (2014). The effects of teaching programming via Scratch on problem solving skills: A discussion from learners' perspective. Informatics in Education-An International Journal, (Vol13_1), 33-50.