CircuitsMaster: An Online End-User Development Environment for IoT Electronics

Ward Seetsen[1], Irene Mavrommati[2], and Vassilis-Javed Khan[3]

[1] CircuitsMaster Eindhoven, The Netherlands

[2] Computer Technology Institute and Press "Diophantus" / Hellenic Open University, Greece

[3] Eindhoven University of Technology, The Netherlands

wardseetsen@circuitsmaster.com

mavrommati@eap.gr

v.j.khan@tue.nl

Abstract. Even though Arduino has made creating products with electronics more accessible, a significant number of users still have difficulties with it. An online tool CircuitsMaster.com (CM), aiming make the design of electronics with Arduino faster and easier is presented in this paper. Three diverse needs of designers that wish to include electronics with Arduino in their projects are explained. CM uses a combination of end-user development paradigms to answer to these needs. Based on results from a user evaluation, those who used CM were significantly faster in creating typical electronics assignments when compared to subjects who did not use CM. Therefore, such tools seem to have a salient role to play for designers that wish to develop IoT products.

Keywords: IoT, End-user development, Arduino, CircuitsMaster, Electronics,

1. Introduction

Creative people come up with ideas, and incrementally improve on those ideas to an innovative concept that needs to be prototypes. In this ideation process, at a point in time designers need to realize a representation of their concept. This takes the form of a prototype, and is an essential part of the design process of realizing ideas. If the concept involves electronics, the prototype will also need to include electronics. Nevertheless, coming up with a great idea involving electronics, does not necessarily mean the person that came up with that idea has the know-how of electronics. Even if one has the know-how the issue of time and effort spent in realizing the concept comes up. Currently the importance of electronics in the design area is gaining significance, due to the rise of the Internet of Things (IoT) (Barricelli & Valtolina, 2015), applications.

In order to make IoT prototypes designers often use Arduino (Fogli et al, 2015). Arduino is an open-source electronics platform that is created to behave as the ’brain’ of any electronic system.
In Arduino, different circuits with different purposes can be connected, for example a motor, a Bluetooth module etc. This is partly made easier due to companies like Adafruit and Sparkfun that sell pre-made circuits, containing all the electronic components for a specific function. When it comes to programming Arduino, one can do that, with a variant of a C-based programming language. However, programming an Arduino still requires some specialized programming skills (Barricelli & Valtolina, 2015). To make Arduino programming easier, several applications are already created; examples include: Ardublock, miniBloq, S4a, Modkit and Visuino. All these platforms have already made prototyping electronics for designers more accessible, there still a diversity of users’ needs that are left unmet, because these platforms can still be experienced as confusing (Barricelli & Valtolina, 2015).
This paper contributes to existing literature regarding programming with Arduino, by presenting:
1) the diversity of design student needs when it comes to programming electronics based an interview study with nine industrial design students
2) CircuitsMaster.com (CM) -a novel online tool for developing electronics, and
3) an experiment that shows that industrial design students CM can develop faster electronic circuits with CM when compared to their current practices.

2. Related Work

Prior research has shown that dealing with electronics is a serious bottleneck for designers (Fogli et al, 2015), (Barricelli & Valtolina, 2015). A series of interviews done with Industrial Design students were done prior to this study to find these bottlenecks. The main bottlenecks were not knowing how to connect the different components, not knowing how to program for Arduino and not understanding the examples that can be found on the Internet. This appeared to cause frustrations for the students.
Prior research to end user programming for applications for Internet of Things has a history since the early 2000’s: Composition of IoT applications can be done interactively, with end-users explicitly involved in the IoT application configuration with the mediation of End User Development (EUD) tools, or automated by intelligent agents (Davidyuk et al. 2015), (Markopoulos et al, 2017). In the latter case (intelligent automation) user involvement is limited, as the system autonomously creates the application’s functionality, although there can be limitations regarding the degree to which user expectations are met.

Programming by Example is one strategy used in End User Development (Dey et al. 2004; Chen and Li 2017). The user in in this approach performs specific examples of system behavior and interaction, and the system infers the application logic. Intelligent systems further support generalization and reuse of the same application logic, by inferring patterns in other similar occasions (Chin et al. 2006). Nevertheless, inferring the correct control logic can be a challenge in programming by example strategies, thus pointing at a need for tools targeted to end users, with which that can further detail application parameters and specify rules. To this end, Rule-based programming is often suggested, with (Huang and Cakmak 2015; Ghiani et al. 2017) proposing a Trigger-action programming style, based on event-condition-action rules.

In other approaches, people compose IoT applications interactively, assisted by Tools that facilitate End User Development. Some such tools use metaphors that can map programming constructs to physical/tangible concepts that users know and can apply more easily. Various such metaphors have been recommended in the relevant literature such as join the dots, pipeline, jigsaw puzzle, (Gross & Marquardt 2007) (Danado and Paternò 2015; Davidyuk et al. 2015). (Fogli et al, 2016). (Kameas and Mavrommati, 2005) proposed an editing tool promoting a high-level conceptual model -a simple visual representation of a “join-the-dots” metaphor- (Mavrommati et al, 2004), while further rule-based editing enables detailed programming of the parameters of the application.

Smart environments created by putting together off-the-shelf smart devices is discussed in (Kubitza and Schmidt 2015). Users can create new applications by using existing devices, concentrating more on the implementation of the application logic. Application’s behavior can be changed with tools that add new context aware rules. Such as a web-based tool has been designed for managing the different application components, and expresses the application logic, such as conditions and events, with JavaScript. This tool seems therefore more fit to be used by programmers rather to other, less computer savy end-users.

The conventional way for a user to instruct Arduino, is a programming language. Yet programming languages are primarily targeting software engineers and not designers. The field of End-User Development wishes to diversify programming to other user groups than professionals. EUD is defined as "A set of methods, techniques, and tools that allow users of software systems, who are acting as non- professional software developers, at some point to create, modify or extend a software artefact" (Barricelli & Valtolina, 2015). According to this definition, CM can be defined as an end-user development tool in the area of electronics where designers are the end-users. The use of end-user development where designers are the end-users is not a novel idea. Several applications have already tried to achieve this goal in other areas of expertise and succeeded. For example, WordPress has allowed people other than web-developers to develop websites. This is widely used by designers (About WordPress, 2010). Other online platforms like Appsbuilder (apps-builder.com) have also achieved this goal for the development of mobile apps.

Several well-researched programming paradigms of end-user development already exist, as mentioned earlier in this section. Programming by demonstration; visual

programming; programming by instruction; and programming by example are the four categories mentioned by (Barricelli & Valtolina, 2015).
In programming by demonstration, the user demonstrates how the software should behave. The problem with this paradigm is that many conditions can easily be forgotten (Schmidt, 2015), especially for complex applications. This causes programs not to behave as desired. Visual programming is used in most of the currently available end-user development Arduino applications. The user is still programming, but with blocks instead of text. This minimizes the errors, but still works from an engineer's point of view and not from a designer's point of view, since the visual blocks are a mere different representation of the programming commands. In programing by instruction, the user inputs rules, instead of programming commands, which describe the desired behavior. An example of this paradigm, is the "IF-THIS-THEN-THAT" (Barricelli & Valtolina, 2015) (ifttt website). It is known as easy to learn (Lucci & Paterno 2015), making this paradigm suitable for CM. In programming by example, the user chooses an example, which closely resembles the desired behavior, and adjusts this example to achieve the exact desired behavior (Lucci & Paterno 2015). Programming by example is a also a known way to create complex applications with little required knowledge, making this also applicable for CM. Wherever programming by example is used, it is important to structure the possible options into intuitive logical categories for users (Lucci & Paterno 2015). So, although several well-known paradigms are used in different domains, when it comes to programming IoT products, Arduino is still the state-of-the-art. This research has set off to investigate whether current designers have unmet needs with Arduino and how can we apply well known EUD methods to diversify design of electronic circuits.

3. Assessment

3.1 Preliminary Study: Interview

Before developing CM, nine industrial students were interviewed: five females and four males, ages between 18 and 25, from two different universities (eight from one department, one from another). Interview questions included, among others: current practices and tools they use to develop electronics; how often they create electronics; how long and what information it takes to create them; their frustrations with current tools.

3.2 Evaluation Experiment

Sixteen industrial design students were recruited -14 from one University and two from another. Their ages ranged from 18 to 26. Their self-reported level of expertise in electronics varied from novice to skilled. One participant was a master student and the others were bachelors, divided over all three bachelor years.
The two research questions were:

a) Are designers' success rate in designing electronics higher with CM than without it?

b) Can designers create electronic circuits faster with CM than without it?

Although we recruited two different groups of participants all of them follow the same educational program -i.e. industrial design. Furthermore, we chose different groups for practical purposes (i.e. to speed up recruitment) as well as avoid potential biases of the first group we interviewed.

3.3 Application and Workflow used

The version of CM that they used for the experiment is further described in this section. CM is unique in that based on IF-THEN rules it automatically generates:

- the Arduino circuit in a graphical representation [Fig.2],
- the list of components that the user needs [Fig.3], and
- the necessary Arduino code
- Furthermore, based on the list of components, CM recommends direct links to e-shops that one could purchase those components

The CM workflow is as follows: the user starts by adding 'objects': these are electronic input and output devices i.e. sensors and actuators. For every chosen 'object' an example circuit is added to the schematic. This way the user ends up with a full schematic for all the required parts. The circuit is build around an Arduino. The user can add *what* the 'objects' should do and *when* these objects should do this i.e. actions and conditions for each action. By setting the actions and conditions for each object, the user gives instructions to CM in an IF-THIS-THAN-THAT way combined with answering the questions CM asks to understand in more detail what the user wants the electronics to do (programming-by-instruction). CM interprets this input and translates it to a combination of code examples to generate a full Arduino code for the user (programming-by-example).

To illustrate the workflow, the following example is used: If the user wants to change an LED color based on the environment temperature, the user first adds an LED. Then the user adds two actions to this LED: turn red and turn blue. Then the user adds a condition to both actions. To the 'turn red' action the user adds the condition 'if the temperature is higher than 20 degrees' and to the 'turn blue' action the user adds the condition 'if the temperature is lower than 20 degrees' [Fig 1]. Then the user clicks 'Generate Circuit' and (s)he receives a circuit with an LED, a temperature sensor and an Arduino code to change the LED color depending on the temperature.

CM is currently available at CircuitsMaster.com. The current version includes 21 components and the premium version includes 50 components. The demo-version used in this experiment is available at http://circuitsmaster.com/Demo/.

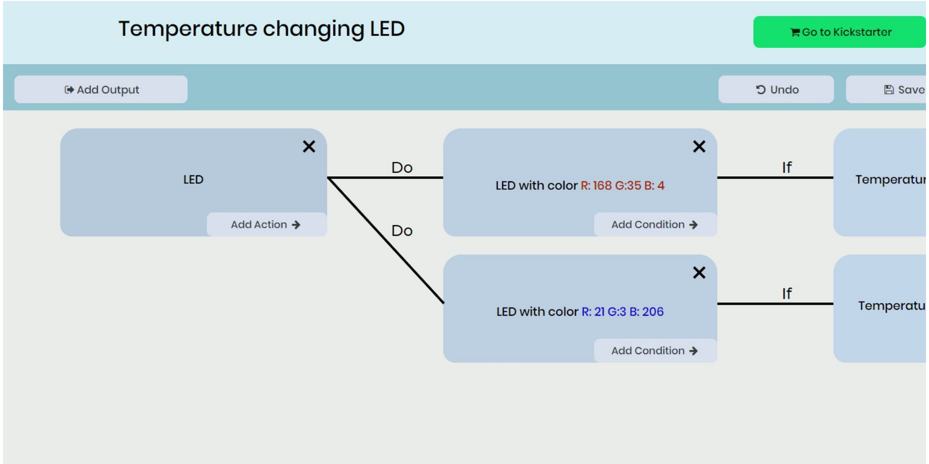


Fig. 1. Circuit design CircuitsMaster

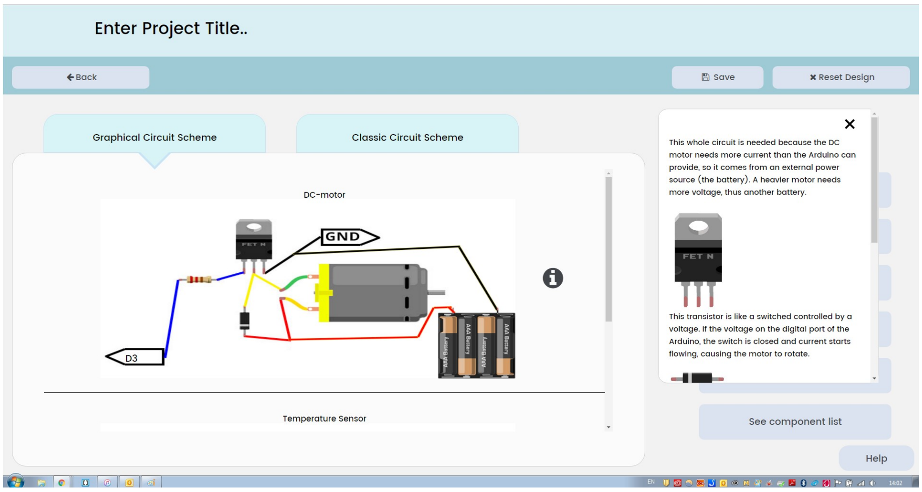


Fig. 2. Circuit generation CircuitsMaster

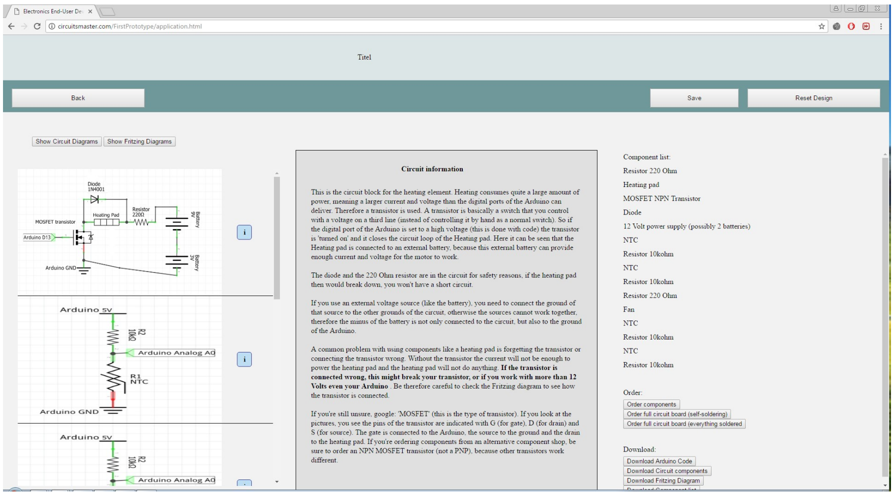


Fig. 3. Screenshot from CircuitsMaster: Easy assignment, output

3.4 Evaluation Process

Initially, participants' skill in electronics were assessed, by asking subjects to grade their skill level and describe about what they had achieved with electronics so far. Based on their answers a participant was classified in the following three categories: novice, intermediate and advanced. Special attention was taken so that in the experiment there was no participation of any experts since they are not CM's target group.

Depending on their skills category, a participant was given an assignment in their level (i.e. easy, intermediate and complex). The easy assignment was to make a thermostat. The intermediate to create an environment system that measured temperature, CO gas and methane gas and indicates if the values are safe. The complex assignment was to create a wake up alarm clock that moves away from the user once it goes off. The difficulty was based on the number of components required and the number of connections between the different components. Participants were asked to complete the assignment up to the point that one could start building it. This in effect means having the circuit scheme, the code and knowing which components to use. A between-subjects research design was opted for -i.e. one group of participants used CM and the other was asked to use whatever method they were used to. Ten participants used CM (3 had novice skills, 4 intermediate, 3 advanced). Six did not use CM and relied to whatever tools they were used to (3 intermediate, 3 advanced). Each participant had 20 minutes to complete the assignment. If the assignment was not completed, participants were asked to estimate the time it would take to finish it. The participant's relative success (in percentage) was also noted when finished. If the participant perfectly completed the assignment, it was assigned a 100%. Otherwise the amount of correct outputs, actions of outputs and inputs were counted and divided by the total amount of outputs, actions and inputs of the correct results. Incorrect actions, inputs or outputs were subtracted from the correct ones. This calculated value then represented the assignment's success percentage for a certain participant. The same method was used for both groups, participants who used CM and those who did not.

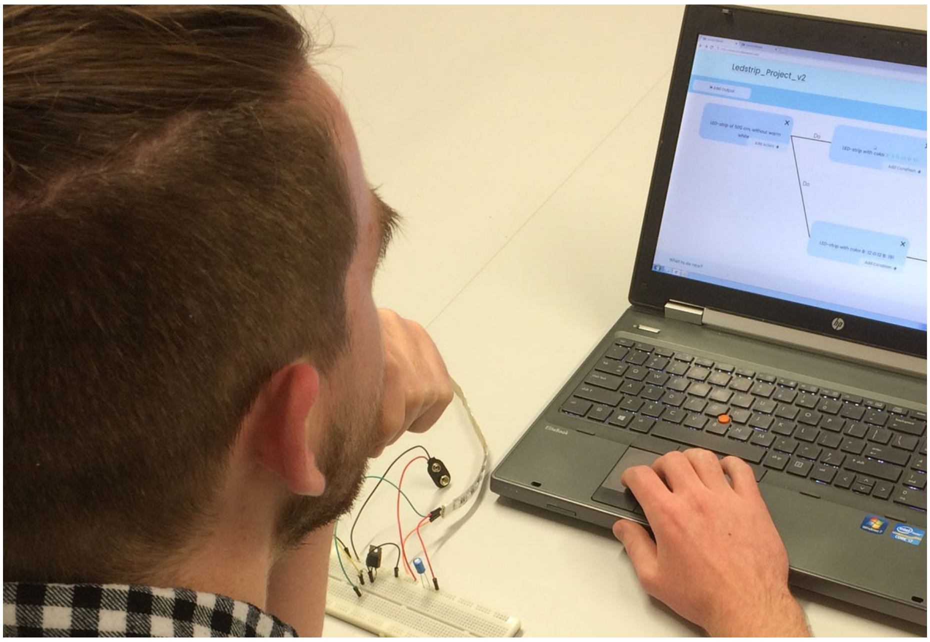


Fig.3. User using CircuitsMaster

4. Results

4.1 Interviews: Three Diverse Unmet Needs

All test participants agreed on the importance of designers being able to design electronics. They need to know what is available regarding electronics and how to use them, because they currently need them in almost every project. Some are really good in using electronics and some are not, there is a considerable variation it this respect. Similarly, there is a variation in appreciating working with electronics or not; some designers like designing the electronics and some do not. However, the participants agreed that knowledge of electronics should not limit their creativity in design. All participating subjects use Arduino in their student projects and build the rest of the electronics around this platform. The main problems they experience is *programming* the Arduino, but also making the *circuit scheme*. An interesting finding is that making the circuit scheme is generally perceived as harder than programming. Another problem that is less mentioned is *soldering*. It is particular frustrating when the electronics break while soldering. Furthermore, getting all *electronics compact* *together* can also be a problem for designers. The estimated time it takes to make the electronics also varies from 15% to 30% of the total project time. After presenting the common findings across all participants, three diverse needs were identified, that are currently unmet for industrial design students. These needs are described by referring to them as "personas".

Persona 1: Uncertain and Afraid.
This persona does not feel confident in making electronics and is afraid to break components. Understanding electronics and different components is a problem and also making a circuit scheme and programming the code. Because of this the use of electronics is generally avoided. The use of electronics in prototypes is experienced as frustrating. CM supports this persona by offering a short explanation for all the circuits and offering the guarantee that every circuit in CM is tested and will work.

Persona 2: Lack of Info.
This persona has more knowledge of electronics, but not enough to create the electronics wanted in prototypes. Everything is created from examples that are adjusted afterwards. The motivation to learn in electronics is there, but there is no time for this, the electronics need to be used now. If necessary, the concept is slightly changed and compromised on the spot so that this persona will be able to implement it. This persona is supported by CM, because all the information this persona needs is in one place: the CM-environment.

Persona 3: Get Results Faster
This persona can already make everything she wants. Also examples are used as the basis of all electronics designs and these examples are adjusted to her personal needs. Values of components are not calculated but she does a calculated guess and afterwards measurements to check if the behavior is as desired (the same holds for sensors and actuators). If necessary, the concept is slightly changed to implement it. Although she can get everything working she gets frustrated if it takes too long for actually creating the product. This persona is supported by CM by all the component-values that are already calculated and by pre-made code that CM provides.

4.2 Experiment: CM Speeds up Development

Due to the fact that the Levene's test for equality of variances was statistically significant for both the success rate and the completion time our data's normal distribution cannot be assumed and therefore a Mann-Whitney test was conducted instead of a t-test. The Mann-Whitney test indicated that the success rate of an assignment is higher with CM (M=93.10%, SD=10.77%, Mean Rank=10.10) than without it (M=55.83%, SD=40.58%, Mean Rank=5.83), but it is not statistically significant U=14, p=.06. The Mann-Whitney test indicated that the time needed for an assignment with CircuitsMaster (M=6.9 min, SD=2.37, Mean Rank=5.5) is lower than the time needed for an assignment without CircuitsMaster (M=189.17 min, SD=150.94, Mean Rank=13.5) and this case the difference is statistically significant

U=0, p<.0001-that is including the extra time that participants had estimated beyond the 20 minutes that we had set for the assignment. The same result is showing if we do not take into account the extra time since all of the 6 participants that did not use CM needed more than 20 minutes to complete the assignment that they were given.

It was also checked whether the three novice users, who were in the CM group, had an effect in the results. The answer to that check is negative. When filtering out those participants the results are the same. Again success rate (Fig.4) was higher for the CM group (M=95%, SD=11.18%, Mean Rank=8.71) when compared to the other group (M=55.83%, SD=40.58%, Mean Rank=5) but not statistically significant U=14, p=.06. Again completion time was faster with CM (M=6.57 min, SD=1.61, Mean Rank=4) when compared to the other group (M=189.17 min, SD=150.94, Mean Rank=10.5) and was statistically significant U=0, p=.001.

Apart from this quantitative data, qualitative data were collected with a debriefing interview. The majority of our participants reported to have enjoyed using the application and stated that it could help them with electronics design.

Some participants also indicated that CM could stimulate their creativity, because it allows them to browse through the possibilities. Participants also stated that the graphical circuit diagrams, made with the Fritzing application (Kraűnig, 2009) were preferred over the classic circuit diagrams. Furthermore, when looking at the actual components that our participants used, we observed that participants using CM were more likely to choose components that can be experienced as complex.

For example, all participants that used CM included a LED strip or neopixel ring (Fig.6,7), while all participants who did not use CircuitsMaster included single LEDs.

An improvement point that was mentioned was about the graphics of CM – the need was pointed for the User Interface design to become more professional. Examples include text fields being selected once they appear, shortcuts and being able to drag-and-drop content.

The participants had divergent opinions about CM’s workflow. Some participants preferred to start with the conditions and work to the output and some participants preferred it the other way around. Also some participants indicated that they would prefer a drag- and-drop interface. Even though participants had divergent opinions on this topic, they all got used to the application quickly.

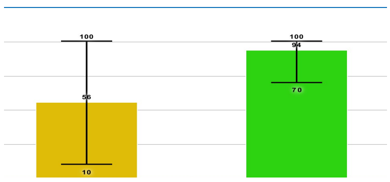


Fig.4. Success Rate with and without CircuitsMaster

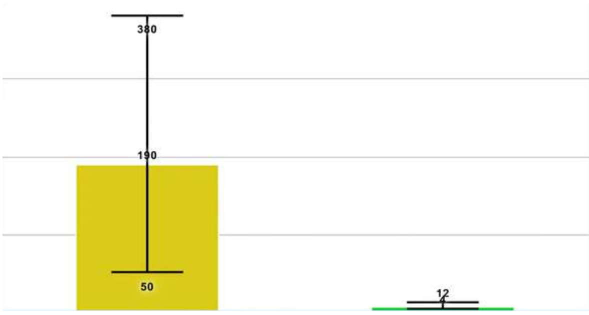
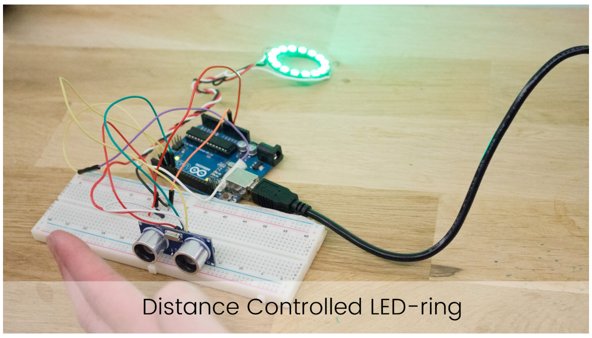


Fig.5. Time in minutes required for assignment with and without CircuitsMaster



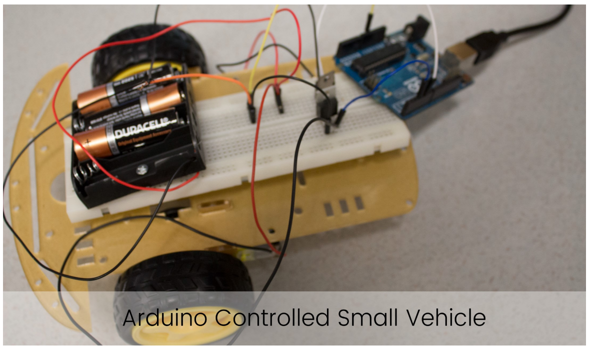


Fig.6,7. Resemblance of projects made during the final user test

5. Discussion - Future Work

The goal of research presented here was to build a tool to help an application to help creative individuals in configuring the electronics in their prototypes. The user tests have shown that CircuitsMaster achieves this in multiple ways, both in succeeding percentage as well as in the amount of time necessary to do these projects. CircuitsMaster is a webtool that has the potential to be added to the toolkit of many designers to help them to make prototypes better and faster. CM as a tool is therefore seen as being a valuable addition to the available tools for electronics design for non-experts. The development of CircuitsMaster will continue since many people can have benefit from this application and a kickstarter campaign has been launched.

There have been limitations of the performed studies: More subjects would be needed so as to draw more valid significant conclusions. One additional limitation is that the first study has been performed with an early prototype of CircuitsMaster. This was done, because the qualitative results of the study were used to improve the prototype and no other prototype was available at the time of the study. It is assumed that the results would be different (improved) with the updated later version, but this assumption has not yet been proven by tests. Repeating the studies with the final version of CircuitsMaster would be a benefit.

The demographics of the participants have been limited. Only results from Industrial Design students from the Technical University of Eindhoven are taken into account in the analysis. This is the case, because the initial target group was Industrial Designers and CircuitsMaster was founded in an Industrial Design Department of the Eindhoven University of Technology. Since the target group Extends further to these profiles, it would be beneficial to redo the full research with the full target group in various locations. Including the full target group means also including participants completely novice in creating electronics, including hobbyist.

From the user studies, several conclusions regarding CircuitsMaster can be drawn: The first one is that users are likely to have a higher success rate when using CircuitsMaster compared to not using CircuitsMaster. Also users can realize their electronics projects faster with CircuitsMaster. Finally, CircuitsMaster is easy to use. CircuitsMaster has also shown that it is possible to generate electrical systems from an input method that is not directly related to the programming language (Fig.8).

The main author continues working with CircuitsMaster to bring it to a point where it can be used and can live up to the expectations of the target group and to the potential of the application. With the help of the crowdfunding campaign, CircuitsMaster aims to be further realized. Once the application is finished, the next phase will be to improve it and explore the use in the several target groups. In the course of this research it was in its preliminary phase and tested only with Industrial Designers, yet, in later stages the target group can be extended. Testing with different target groups will be able happen after the finalization of the first version to be launched.

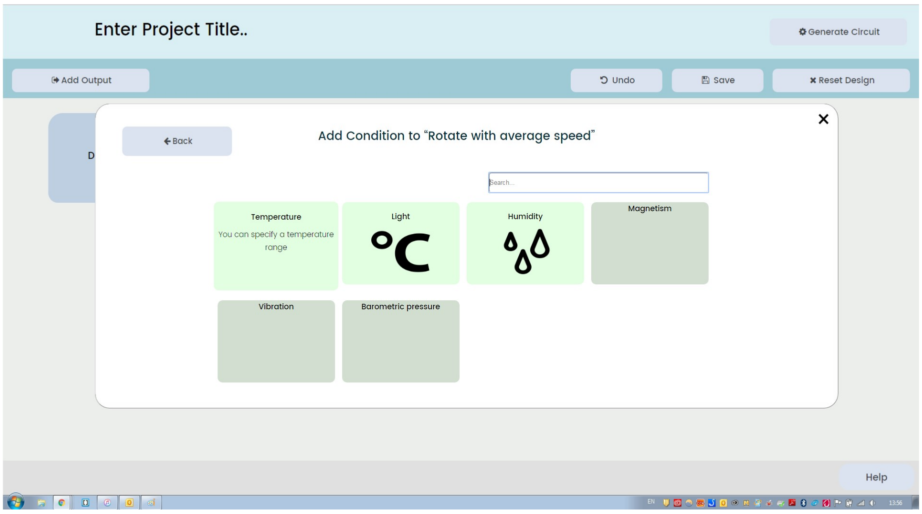


Fig. 8. Instructing CircuitsMaster

6. Conclusions

While recognizing the limited number of participants in this study, two important contributions need to be stressed, that pave the way for future research studies. The first contribution considers the diversity of unmet needs that industrial designers have with regards to designing products with electronics. Based on an interview study (N=9) we identify three diverse profiles of designers that future electronics tools need to support designers in:

1) boosting the confidence, especially of novice ones, by clarifying what components are needed and how to connect them;
2) improving their knowledge by integrating examples they can consult relating to the specific project they are currently working one;
3) making the creation of the circuit faster.

The second contribution considers CircuitsMaster.com (CM), an online tool that is designed to support the three aforementioned needs. In an in-between subjects' experiment with industrial design students (N=16) we showed that CM speeds up considerably the completion time of typical assignments, with different levels of difficulties, that involve electronics.

Furthermore, although not statistically significant, participants who used CM had a higher success rate (M=93.10%, SD=10.77%) within a 20 minutes' window when compared to students that did not use CM (M=55.83%, SD=40.58%). These results make us confident that with further improvement CM can become a salient weapon in the arsenal of industrial designers that wish to develop IoT products. Future work includes developing further CircuitsMaster.com. Apart from adding more components, CM will support sharing projects among users, and active guidance in writing code and in actually building the circuit.

References

7. Fogli, D., Giaccardi, E., Acerbis, A., & Filisetti, F. (2015). Physical Prototyping of Social Products Through End-User Development. End-User Development Lecture Notes in Computer Science, 217-222. doi:10.1007978-3319184258_19
8. Barricelli, B. R.& Valtolina, S. (2015). Designing for End-User Development in the Internet of Things. End-User Development Lecture Notes in Computer Science, 924. doi:10.10079783319184258_2
9. Adafruit https://www.adafruit.com/ , last accessed July 2019
10. Sparkfun https://www.sparkfun.com/ last accessed July 2019
11. Davidyuk O, Milara IS, Gilman E, Riekki J (2015) An overview of interactive application composition approaches. *Open Computer Science*, *5*(1).
12. Markopoulos P, Nichols J, Paternò F, Pipek V (2017) End-User Development for the Internet of Things. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 9.
13. Dey AK, Hamid R, Beckmann C, Li I, Hsu D (2004) a CAPpella: programming by demonstration of context-aware applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (pp. 33-40). ACM.
14. Chen XA, Li Y (2017) Improv: an input framework for improvising cross-device interaction by demonstration. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 15.
15. Chin JSY, Callaghan V, Clarke G (2006) An End-User Programming Paradigm for Pervasive Computing Applications. In *ICPS* (Vol. 6, pp. 325-328).
16. Huang J, Cakmak M (2015) Supporting mental model accuracy in trigger-action programming. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (pp. 215-225).
17. Ghiani G, Manca M, Paternò F, Santoro C (2017) Personalization of context-dependent applications through trigger-action rules. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *24*(2), 14.
18. Gross T, Marquardt N (2007) CollaborationBus: An editor for the easy configuration of ubiquitous computing environments. In *Parallel, Distributed and Network-Based Processing, 2007. PDP'07. 15th EUROMICRO International Conference on* (pp. 307-314). IEEE
19. Danado J, Paternò F (2015) A Mobile End-User Development Environment for IoT Applications Exploiting the Puzzle Metaphor. *ERCIM News*, *101*, 26-27.
20. Fogli D, Lanzilotti R, Piccinno A (2016, July) End-user development tools for the smart home: a systematic literature review. In *Proceedings of the International Conference on Distributed, Ambient, and Pervasive Interactions* (pp. 69-79). Springer, Cham.
21. Kameas A, Mavrommati I (2005) Extrovert gadgets. *Communications of the ACM*, *48*(3), 69.
22. Mavrommati I., Kameas A., Markopoulos P. An editing tool that manages device associations in an in-home environment. Personal and Ubiquitous Computing 8 (3-4), 255-263
23. Kubitza T, Schmidt A (2015) Towards a toolkit for the rapid creation of smart environments. In *International Symposium on End User Development* (pp. 230-235). Springer, Cham.
24. About WordPress. (2010). Beginning WordPress 3, 1-14. doi:10.1007978-1430228967_1
25. apps-builder.com: Create & publish powerful native apps. (n.d.). Retrieved February 06, 2017, from http://www.apps-builder.com/

26. Schmidt, A. (2015). Programming Ubiquitous Computing Environments. End-User Development Lecture Notes in Computer Science, 36. doi:10.10079783319184258_1
27. Lucci, G., & Paterno, F. (2015). Analysing How Users Prefer to Model Contextual Event-Action Behaviours in Their Smartphones. End-User Development Lecture Notes in Computer Science, 186191. doi:10.10079783319184258_14
28. Kraűnig, A. (2009). Fritzing: A tool for advancing electronic prototyping for designers.
29. ifttt.com If This Then That: a free web-based service to create chains of simple conditional statements, called applets.
30.