Light Alarm

Intermediate Version

Georgios Mavromanolakis

[gmavroma@ea.gr]

Stimulate problem solving	**Silver**	**Stimulate entrepreneurship**	**Silver**
Stimulate creativity	**Silver**	**Informal learning enviro.**	**Silver**
Stimulate critical thinking	**Silver**	**Technology use**	**Gold**
Stimulate group work	**Silver**		

Practicalities

Preparation: 10min

Duration: 2hrs

Material needs:

- Arduino UNO, 9V battery, jumper wires, breadboard, photoresistor, buzzer

Group size range: 15-20

Ideal sub-group size: 2-3

Workshop made for: 16+ years old
Easily transferable to workshops for ages between: 12-16

Environment FabLab necessary: No

Educational area:
* Computer science
* Engineering
* Science
* Technology

Precognition

Basic coding/programming skills are needed. Familiarization working with Arduino platform is required.

Preparation

Students work individually or in small groups of 2 to 3 and their task is to make a functional light alarm device using an Arduino board, photoresistor and buzzer.

In the preparation phase, teachers/educators collect and sort the materials. Each student team should have the following materials:

An Arduino UNO board (see pic below)
A photoresistor (see pic below)
A buzzer (see pic below)
A bundle of jumper wires (and optional a breadboard)
(Optional) A schematic of the circuit to build, attached at the end of this document
(Optional) On paper, the source code to run on Arduino, attached at the end of this document

Workshop Guidelines

Phase 1: Orientation and Instruction Phase

Material needs:
As described above

Goals:
Skill Goals (**Blue**)
(S1) To look up information independently

Content Goals (**Green**)
(C1) To understand a schematic of a circuit
(C2) To identify each component and its functionality
(C3) To pseudocode or describe code in natural language, to devise a flow of actions

Background story:

Open source hardware and software is easily accessible to everyone for fun, education, serious applications. In this workshop students make a low-cost, simple but operational, electronic device, a light or darkness alarm, practicing and experiencing basic making and coding skills.

Goals	Activities	Duration
S1	Ask students to look up for basic information about each component they have at hand.	Depending on level of students
C1, C2, C3	Let students to familiarize with the schematic of the circuit and its components. Ask them to describe in natural language what the flow of actions can be so that such device functions. (Optional, instruct them to draw a block diagram).	Same as above

Phase 2+3: Design and Making Phases

Material needs:

Essential: same as above

Goals:

Skill Goals (**Blue**)

(S1) Transfer of a representation (schematic) to a real artifact (circuit)

Content Goals (**Green**)

(C1) To understand a schematic of a circuit

(C2) To identify each component and its functionality

Goals	Activities	Duration
S1, C1, C2	Student teams build the circuit using the given materials. At these phases they do not upload or run any code on Arduino. (Optional, for advanced level of students with acquired coding and making skills, the schematic diagram of the circuit may not be given in advance. They must develop it themselves).	20 min

Phase 4: Operational Phase

Material needs:

Essential: same as above + source code on paper

Goals:

Skill Goals (**Blue**)
(S1) Trial and error / Deal with failure
(S2) (Optional) Competing with others
(S3) Problem solving

Content Goals (**Green**)
(C1) Coding/programming a device (physical computing)
(C2) To identify each element and its functionality in the code

Goals	Activities	Duration
S1, S2, S3, C1, C2	In this phase students first type/transfer the code from paper to the Arduino software. They compile and upload their code on Arduino board. They run it and test if it functions correctly. (Optional, for advanced level of students with acquired coding skills, the source code may not be given in advance. They must develop it themselves). They proceed with corrections and debugging if necessary.	40 min

Phase 5: Evaluation Phase

Material needs:

Essential: same as above

Goals:

Skill Goals (**Blue**)
(S1) Deal with failure
(S2) (Optional) Competing with others
(S3) Problem solving
(S4) (Optional) Acquire entrepreneurial mindset

Content Goals (**Green**)
(C1) Coding/programming a device (physical computing)
(C2) To identify each element and its functionality in the code

Goals	Activities	Duration
S1, S2, C1, C2	In the first round each team demonstrates that its device functions properly as light alarm. It also discusses its possible applications.	20 min
S1, S2, C1, C2	In the second round, the teams are asked/challenged to make their device to function in opposite way, i.e. as darkness alarm. To do this they only have to alter the source code accordingly. They also discuss for possible applications.	20min

Pedagogical tips

The workshop can be enhanced and advanced with multiple components, sensors, actuators etc. and multiple rounds of design-make-evaluate so that teams can try multiple ideas, built various devices, combine and test them. Similarly, it can be made easier and transferred to lower ages of students with the use of visual programming environment like Scratch or other.

How to transfer to (non-)Fablab environment

Transfer to non-fablab environment is very feasible, as long as the necessary materials are provided.

Evaluation of achievements

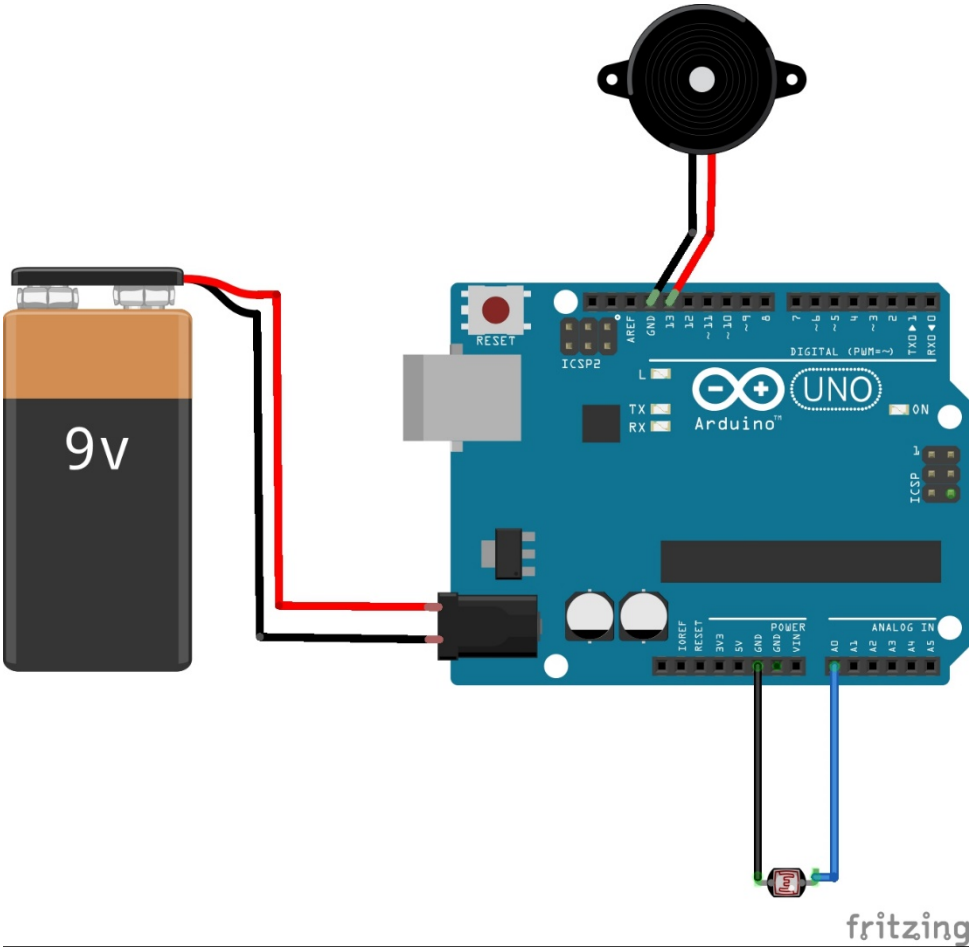
At the end of the workshop you can award different teams depending on achievements.
For example, award for:

- The team who succeeded in less time
- The team who proposed the most innovative application
- The team who proposed the most outrageous/useless application
- ...

Content links

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. It can be downloaded from https://www.arduino.cc/en/Main/Software

Schematic of connections



Source code for Arduino

```
/////////////////////////////////////////////////////
// Connection table
/////////////////////////////////////////////////////
//
// Photoresistor module
//
//    for a 3-pin photoresistor then connect
//    Arduino A0  --> Module Signal (S)
//    Arduino 5V  --> Module +5V (middle Pin)
//    Arduino GND --> Module GND (-)
//
//    for a 2-pin photoresistor then connect
//    Arduino A0  --> Module Signal (S or +)
//    Arduino GND --> Module GND (-)
//
//
// Buzzer module
//    Arduino D13 --> Module pin S or +
//    Arduino GND --> Module pin -
/////////////////////////////////////////////////////

int sensorPin = A0;   // analogue pin to read sensor signal
int sensorValue = 0;  // variable to store signal from sensor
int buzzerPin = 13;   // pin to activate the buzzer

// This is called only once
void setup()
{
  // declare the buzzerPin as an OUTPUT
  pinMode(buzzerPin, OUTPUT);
}

// This called continuously again and again
void loop()
{
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);

  //value is around
  //200-250 with lamps on,
  //350-400 for ambient light,
  //800-900 when sensor is covered by hand
  //
  //check the range of values that are appropriate for your case
```

```
  // turn on buzzer when sensor is not covered
  if(sensorValue < 600) digitalWrite(buzzerPin, HIGH);
  else digitalWrite(buzzerPin, LOW);

  // wait for 10msec
  delay(10);
}
```