PHYSICAL SENSORS FOR ENVIRONMENTAL SIGNALS

Irene Nutini

Master Degree in Artificial Intelligence for Science and Technology (AI4ST)

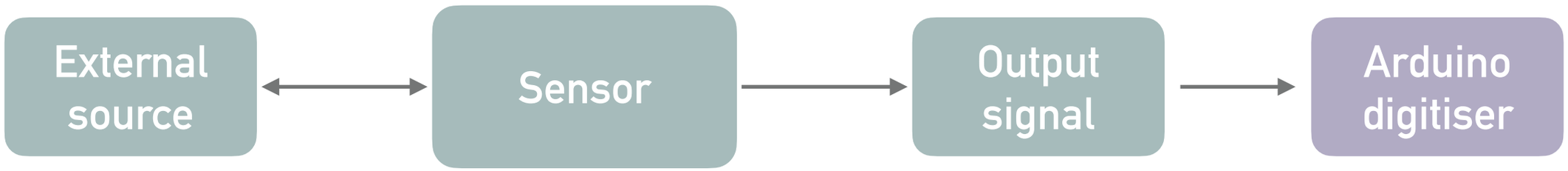
A.y. 2024-2025

OUTLINE OF THE COURSE

- Lecture 1: Introduction to environmental signals and physical sensors
- Lab 1: Introduction to instruments for measurements
- Lecture 2: Vibrations: sources and detection
- Lab 2: Characterisation of an acoustic system
- Lecture 3: Distance, position and speed measurement
- Lab 3: Measuring distance with ultrasounds and speed with an accelerometer
- Lecture 4: Electromagnetic radiation: sources and detection
- Lab 4: Detecting and generating light

SENSING THE ENVIRONMENT

EXAMPLE: LIGHT DETECTION WITH A PHOTOSENSOR



- Source: Ambient light / LED / torch
- Sensor: Phototransistor
- Read the signal output: Arduino digitiser

Lab.4 (today)

DETECTING LIGHT

- Source: Ambient light / LED
- Sensor: Phototransistor
- Read the signal output: Arduino digitiser

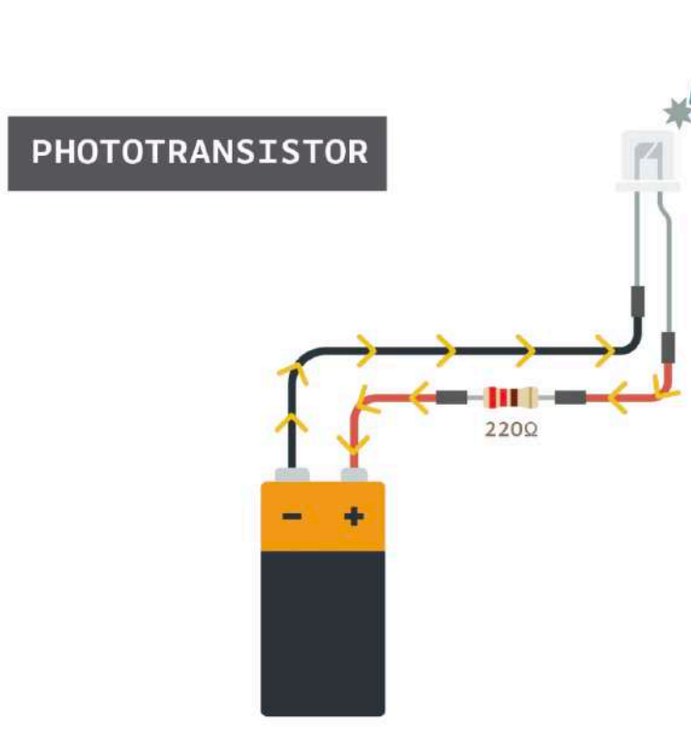
We'll use the Arduino board and a phototransistor to investigate electromagnetic waves – specifically visible light. We'll explore some of the applications of detecting and measuring light such as communication.

Reference: *https://studentkit.arduino.cc/studentkit/module/student-kit/lesson/light-wave-radar*

DETECTING LIGHT

- Phototransistor

A **phototransistor** is an electronic component that converts light energy into electrical energy.



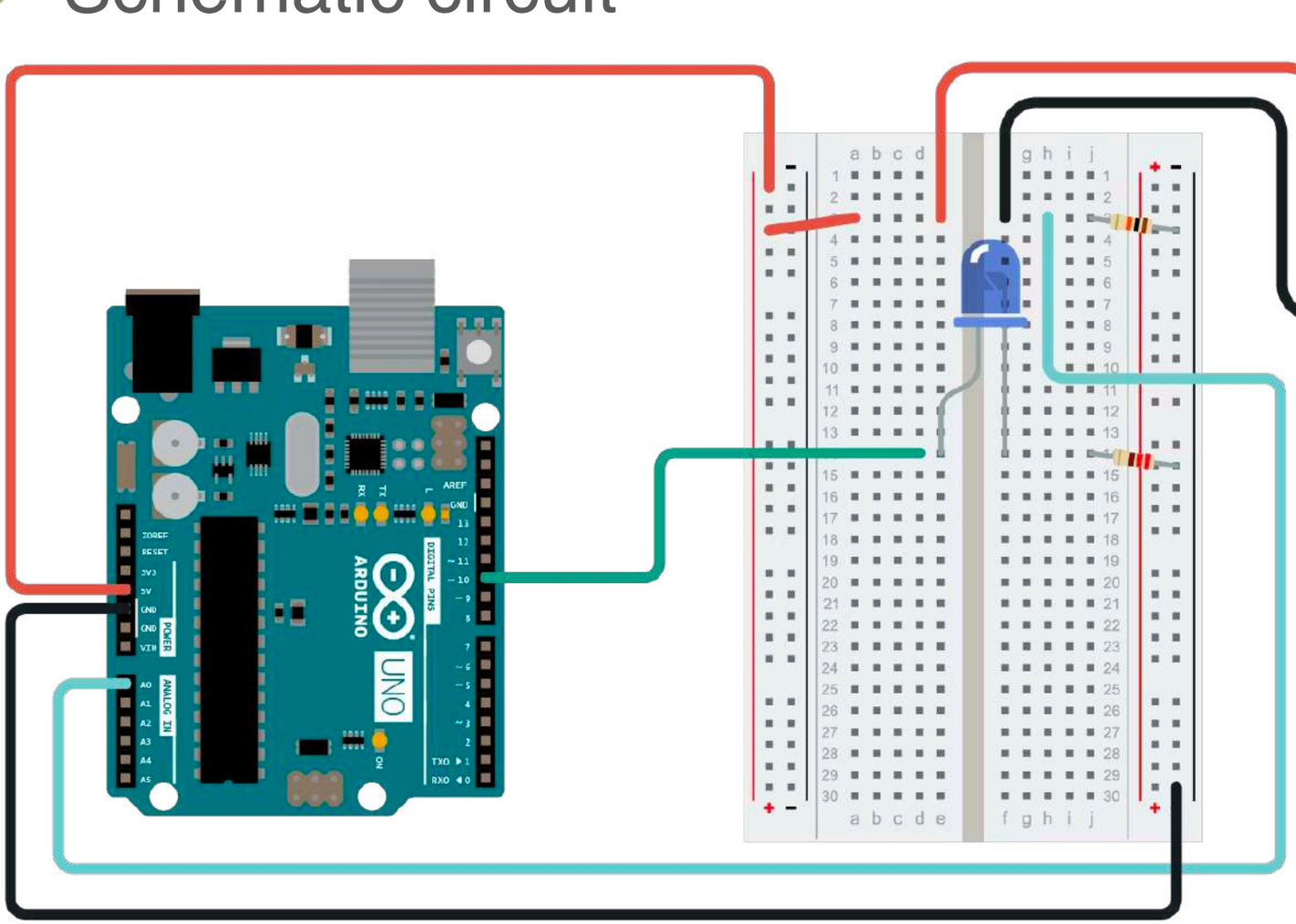
In Lab:

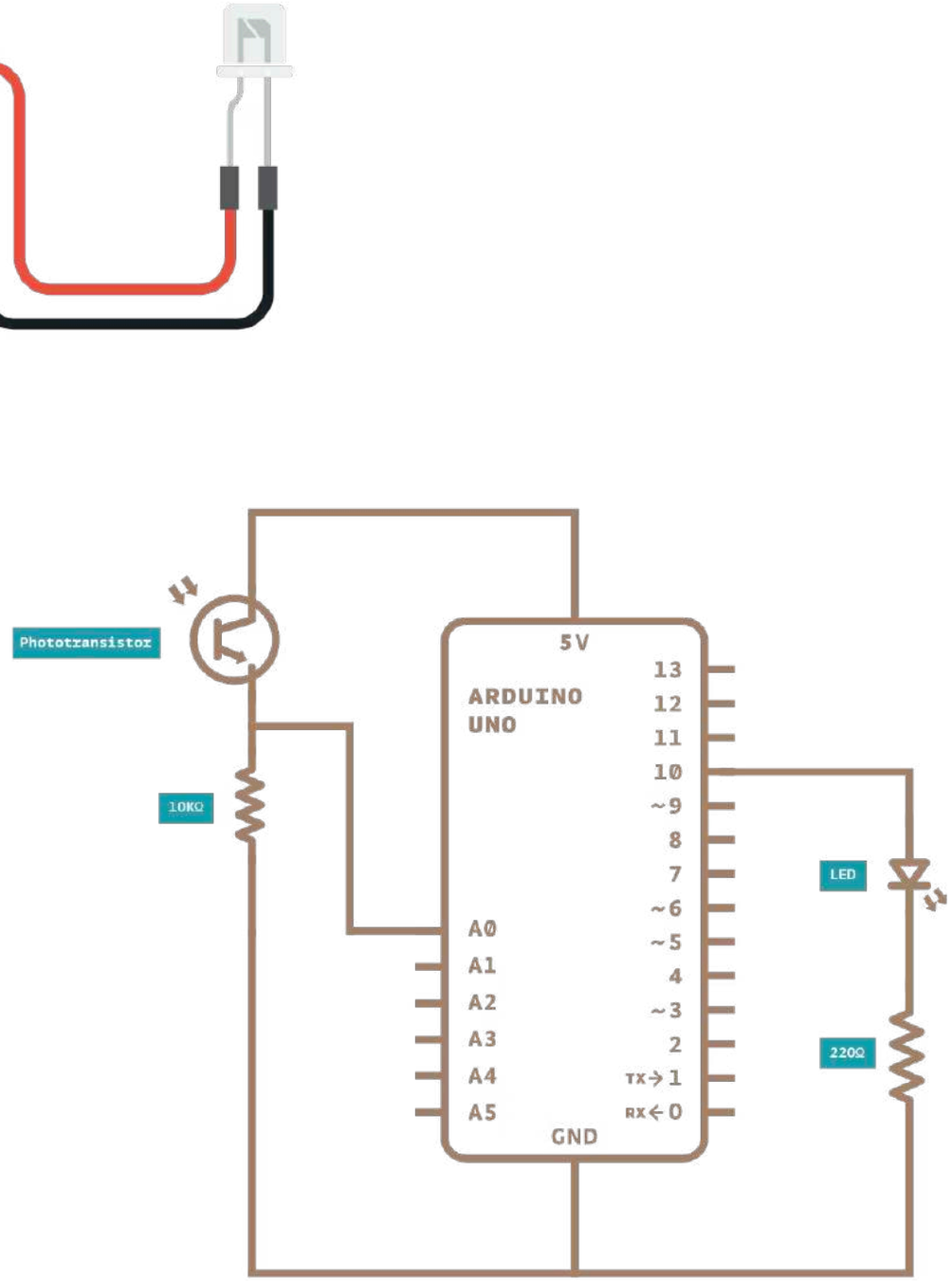
BPW40 Silicon NPN Epitaxial Planar Phototransistor

http://www-9.unipv.it/lde/strumentazione_comp_extra/datasheet/BPW40.pdf

DETECTING LIGHT

- Schematic circuit

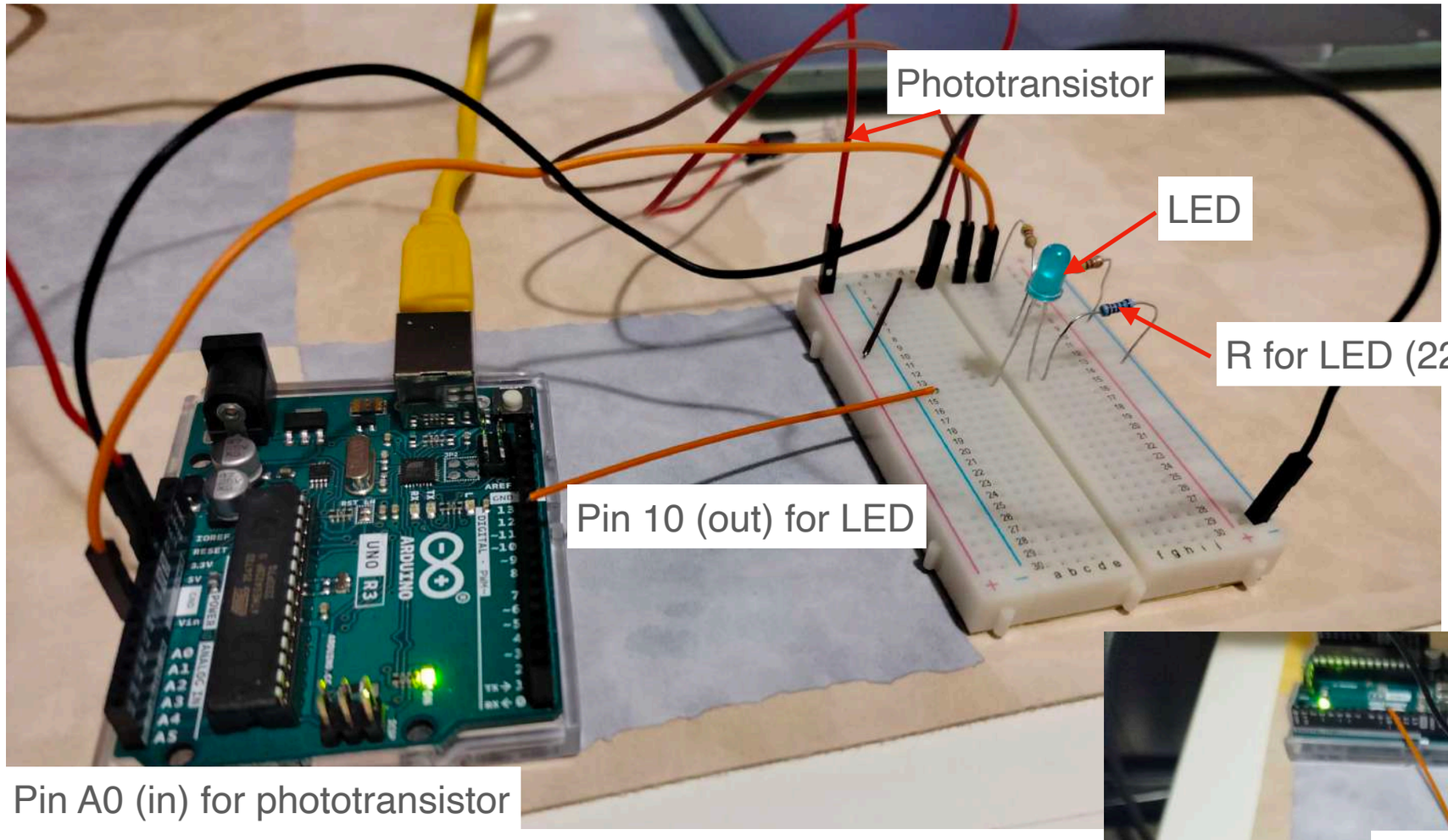


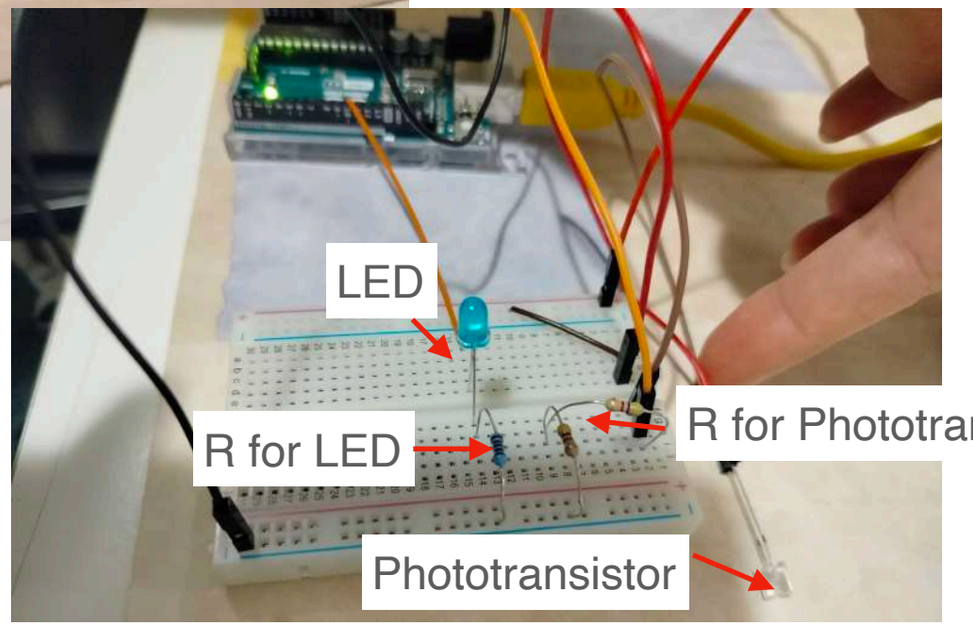


Note: in Lab, red LED

DETECTING LIGHT

- Real circuit





DETECTING LIGHT

- Reading the phototransistor signal with Arduino serial output

1. Use the phototransistor as a light sensor to detect the brightness, or intensity, of the light in the room. (See Arduino sketch —>)

2. Move the light sensor around and observe the measurement on the serial monitor. Cover the light sensor with your hand. Point the light sensor at a light in the room. Try different light sources: smartphone torch, IR/red torch, UV torch…

Photodetector.ino

```
1   // name Arduino board pins used by the circuit
2   const int sensorPin = A0;
3
4   // declare variables
5   int lightAmount = 0;
6
7   void setup() {
8     // put your setup code here, to run once:
9     // start the serial monitor
10    Serial.begin(9600);
11
12  }
13
14  void loop() {
15    // put your main code here, to run repeatedly:
16    lightAmount = analogRead(sensorPin);
17
18    // output info to the serial monitor
19    Serial.print("Light Intensity: ");
20    Serial.println(lightAmount);
21
22    delay(1000);
23  }
24
```

DETECTING LIGHT

- Reading the phototransistor signal with Arduino serial output

1. In this activity, we use an LED and light sensor to simulate sending information through light waves (eg. Optical fiber transmission of information).

2. Modify the previous Arduino sketch to cause the blue LED to blink and act as a light source. As the light sensor is oriented at the LED, we observe how the light sensor reacts to the blinking light.

Photodetector_LED.ino

```
// name Arduino board pins used by the circuit
const int sensorPin = A0;
const int LEDPin = 10;

// declare variables
int lightAmount = 0;
long timerLED = 0;
long timerSensor = 0;
int toggleLED = 0;

void setup() {
  // put your setup code here, to run once:
  pinMode(LEDPin, OUTPUT);
  // start the serial monitor
  Serial.begin(9600);

}

void loop() {
  // put your main code here, to run repeatedly:
  //In the void loop() part of the sketch, you will have two tasks.
  //The first task is to create a transmitter that flashes the light.
  //The second task is to create a receiver that detects the light

  transmitter();
  receiver();

  //delay(1000);
}

void transmitter(){
  // code for transmitting a signal in 4-second intervals
  if (millis() >= timerLED + 2000) {
    toggleLED = !toggleLED;
    digitalWrite(LEDPin, toggleLED);
    timerLED = millis();
  }
}

void receiver(){
  // code for receiving signals every 100 milliseconds
  if (millis() >= timerSensor + 100) {
    // read the light sensor and store the measurement in a variable
    lightAmount = analogRead(sensorPin);

    // output info to the serial monitor
    Serial.print("Light Intensity: ");
    Serial.println(lightAmount);

    timerSensor = millis();
  }
}
```

EXTRA (1): COLLECTING AND GENERATING LIGHT SIGNAL

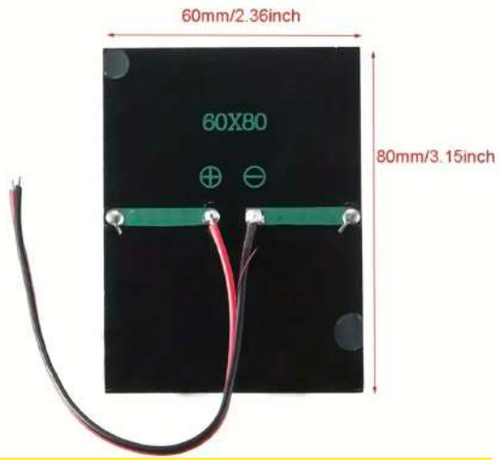
- Collecting light and generating current to turn on LED

1. Collecting light with small solar cell:

   I. Expose the cell to indoor white light, UV torch, IR/red torch

   II. Read the analog voltage output of the solar cell with multimeter* and digitise it with Arduino (print it on serial output)

   III. Compare the analog output of the solar cell for different light sources, distances of the torch and shadowing condition



Solar cell technical details:
Power 0,65 W
Current 0-300 mA
Voltage 1,5 V

*Check both the voltage and current output of the solar cell with the multimeter

Example video: https://youtu.be/nBxEjC2Gzqs?si=fwQW-m_49y2rM_4x

EXTRA (1): COLLECTING AND GENERATING LIGHT SIGNAL

- Collecting light and generating current to turn on LED

2. Collecting light with few small solar cells in series and turn on an LED

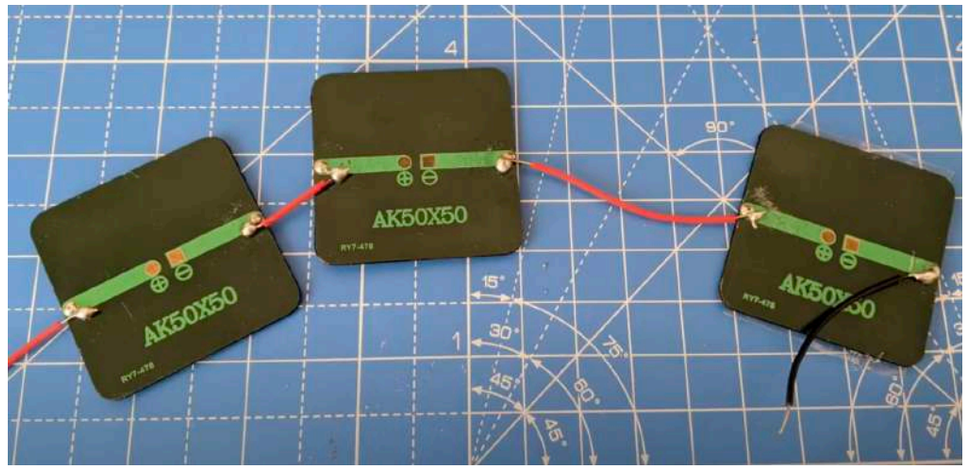
I. Connect 2 solar cells in series

II. Connect a LED (different colors - start with red) to the solar cell voltage output

III. Read the analog voltage output (and the current output) of the solar cell system with multimeter for different exposures to light.

IV. Check for which voltage and light exposure conditions the different LEDs turn on

Example: 3 solar cells in series



Example video: https://www.youtube.com/shorts/4MB7nkSjMro

EXTRA (2): GENERATING LIGHT SIGNAL

- Generating light signal (with morse code) giving serial input to Arduino

1. In this activity, we'll see that the serial monitor can be used to input information and tell the Arduino UNO R3 board what to do.

2. You can turn on and off the LED of the circuit (pin 10) by inputting commands into the serial monitor. On the input line of the serial monitor: H - turns the LED on, L - turns the LED off

3. Send an SOS signal in morse code with LED blinking light by serial input

```
/*
  Physical Pixel - SOS
  From: https://www.arduino.cc/en/Tutorial/BuiltInExamples/PhysicalPixel
*/

const int ledPin = 10;// the pin that the LED is attached to
int incomingByte;       // a variable to read incoming serial data into

void setup() {
  // initialize serial communication:
  Serial.begin(9600);
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
}

void loop() {
  // see if there's incoming serial data:
  if (Serial.available() > 0) {
    // read the oldest byte in the serial buffer:
    incomingByte = Serial.read();
    // if it's a capital H (ASCII 72), turn on the LED:
    if (incomingByte == 'H') {
      digitalWrite(ledPin, HIGH);
      delay(300);
    }
    // if it's an L (ASCII 76) turn off the LED:
    if (incomingByte == 'L') {
      digitalWrite(ledPin, LOW);
      delay(300);
    }
  }
  // SOS
  // HL HL HL HHL HHL HHL HL HL HL
}
```

international morse code

A ● ▬	M ▬ ▬	Y ▬ ● ▬ ▬
B ▬ ● ● ●	N ▬ ●	Z ▬ ▬ ● ●
C ▬ ● ▬ ●	O ▬ ▬ ▬	1 ● ▬ ▬ ▬ ▬
D ▬ ● ●	P ● ▬ ▬ ●	2 ● ● ▬ ▬ ▬
E ●	Q ▬ ▬ ● ▬	3 ● ● ● ▬ ▬
F ● ● ▬ ●	R ● ▬ ●	4 ● ● ● ● ▬
G ▬ ▬ ●	S ● ● ●	5 ● ● ● ● ●
H ● ● ● ●	T ▬	6 ▬ ● ● ● ●
I ● ●	U ● ● ▬	7 ▬ ▬ ● ● ●
J ● ▬ ▬ ▬	V ● ● ● ▬	8 ▬ ▬ ▬ ● ●
K ▬ ● ▬	W ● ▬ ▬	9 ▬ ▬ ▬ ▬ ●
L ● ▬ ● ●	X ▬ ● ● ▬	0 ▬ ▬ ▬ ▬ ▬