**Cyrille Robotic ASS 1 (I wrote it on a jupyter notebook in order to make it more readable for you but it works on Thinkercad. You just need to copy/paste it on thinkercad. I added the links for each exercices so that you can check the code easily in C++. At the end, I added the plots and the circuits)**

EX 1

```
// Cyrille's CODE
//First Part
const int LEDREDpin = 4;//the pin where led red is plugged
const int LEDGREENpin = 8;//the pin where led green is plugged
const int LEDBLUEpin = 13;//the pin where led blue is plugged
const long onDuration = 1000;// ON time for every LED
int LEDREDState =HIGH;// initial state of LED RED
int LEDGREENState =LOW;// initial state of LED GREEN
int LEDBLUEState =LOW;// initial state of LED BLUE

long TimeCounter=0;// this is like a counter that will change during
the process by taking the value of millis. At the start of the
program, its value should be 0

//Second Part
void setup() {//classic synthax in c++
  pinMode(LEDREDpin,OUTPUT);// define LED red pin as output
  digitalWrite(LEDREDpin,LEDREDState);// set initial state for led red
  pinMode(LEDGREENpin,OUTPUT);// define LED green pin as output
  digitalWrite(LEDGREENpin,LEDGREENState);// set initial state for led
green
  pinMode(LEDBLUEpin,OUTPUT);// define LED blue pin as output
  digitalWrite(LEDBLUEpin,LEDBLUEState);// set initial state for led
blue
}

//Third Part
void loop() {//classic synthax in c++ to start my loop

 if( LEDREDState ==HIGH )//the general condition (1)
 {
    if( (millis()- TimeCounter) >= onDuration){//the specific
condition (1)
    LEDREDState = LOW;// the state of LED red when the conditions (1)
are met
    LEDGREENState = LOW;// // the state of LED green when the
conditions (1) are met
    LEDBLUEState = HIGH;// the state of LED blue when the conditions
(1) are met
    TimeCounter=millis();// the counter remember Current millis() time
    }
 }
```

```
  if( LEDBLUEState ==HIGH )// the general condition (2)
  {
    if( (millis()- TimeCounter) >= onDuration){//the specific
condition  (2)
    LEDBLUEState = LOW;// the state of LED blue when the conditions
(2) are met
    LEDREDState = LOW;//the state of LED red when the conditions (2)
are met
    LEDGREENState = HIGH;//the state of LED green when the conditions
(2) are met
    TimeCounter=millis();// the counter remember Current millis() time
    }
 }
 if( LEDGREENState ==HIGH )// the general condition (3)
  {
    if( (millis()- TimeCounter) >= onDuration){//the specific
condition  (3)
    LEDGREENState = LOW;//the state of LED green when the conditions
(3) are met
    LEDREDState = HIGH;//the state of LED red when the conditions (3)
are met
    LEDBLUEState = LOW;//the state of LED blue when the conditions (3)
are met
    TimeCounter=millis();// the counter remember Current millis() time
    }
 }


//Last Part
 digitalWrite(LEDREDpin,LEDREDState);// turn the LED red ON or OFF
 digitalWrite(LEDBLUEpin,LEDBLUEState);// turn the LED blue ON or OFF
 digitalWrite(LEDGREENpin,LEDGREENState);// turn the LED green ON or
OFF
}// the end
```

**SUMMARY**: I used Rajeev's explanations from the robotics class and the tutorials seen on YouTube to set up the device. I had 3 leds with 3 resistors of 220 ohm connected to the Adruino uno R3 with wires and a breadboard. I chose to use different colors to make my work easier to understand. The leds were plugged on ramdomly chosen Digital pins (4, 8, 13) and the Ground pin (GND).

**For the first part of the code**, I just defined some variables so that I can tell my device where the LEDs are connected, how long they should be on and what their starting state is. To answer Rajeev's request, I had to start with one led on and the others off. I also decided to set a time counter that will change its values according to the conditions. **For the Second part of the code**, I used the setup() function. The setup function runs once when you press reset or power the board. Thanks to this function, I initialize the digital pins (where the leds are plugged) as output and I set the initial state of each led. Now My Adruino knows the outputs. **For the Third part of the code**, I used the loop() function. The

loop function runs over and over again forever. With this function, I defined some conditions where the leds must be on or off thanks to the 'if' statement in order to follow Rajeev's request. The different conditions of change concerned the state of the led and the time. For the time, I had to use the millis() built-in function which return the number of milliseconds passed since the program started. In summary, if a led x was on (general condition) and if the 'on duration' was exceeded (specific condition), led x had to turn off, led y had to turn on and led z had to stay off. If a led y was on (general condition) and if the 'on duration' was exceeded (specific condition), led y had to turn off, led z had to turn on and led x had to stay off. And so on... At the end of each specific condition the value of the counter take the value of millis. **For the Last part of the code**, I simply made my leds turning on and off simultaneously (in function of my conditions).

Link for ex 1: https://www.tinkercad.com/things/4q8UXAPfK6b-exercice-1/editel?sharecode=M8X1_oYvemVGlRCYwzcbjLCnkKV3OYLQVfI3UXFiBHk

EX2

```
// Cyrille's CODE
//First Part
int r=12;//the pin where led red is plugged
int g=4;//the pin where led green is plugged
int pot=A0;//the pin where the potentiometer is plugged
int potbuttonval=0;//this is like a counter that will change during
the process by taking the value of the position of the
potentiometer's' Button. At the start of the program, its value should
be 0

//Second Part
void setup()
{
  pinMode(r, OUTPUT);// define LED red pin as output
  pinMode(g, OUTPUT);// define LED green pin as output
  pinMode(pot, INPUT);// define the potentiometer's' pin as input
(because that's' the thing that we will change in order to modify the
brightness of the leds)
}

//Last Part
void loop()
{
  potbuttonval=analogRead(pot);// read the value from the analog pin
where the potentiometer is plugged (here it's' A0)
  analogWrite(r,potbuttonval/4);// turn the LED red ON or OFF.
Dividing potbutton by 4 to bring it in range of 0 - 255
  analogWrite(g,potbuttonval/4);// turn the LED green ON or OFF.
Dividing potbutton by 4 to bring it in range of 0 - 255
  delay (100);//wait for 100 milliseconds
}//the end
```

**SUMMARY**: I used Rajeev's explanations from the robotics class and the tutorials seen on YouTube to set up the device. I had 2 leds with 2 resistors of 220 ohm connected to the Adruino uno R3 with wires and a breadboard. I also added a 10 kilo ohm Potentiometer that will control the leds brightness. I chose to use different colors to make my work easier to understand. The leds were plugged on ramdomly chosen Digital pins and the Ground pin (GND). And the Potentiometer on an analog pin. The power chosen was 5V.

**For the first part of the code**, I just defined some variables so that I can tell my device where the LEDs and the Potentiometer are connected. I also decided to set a counter called 'potbuttonval'. **For the Second part of the code**, I used the setup() function. The setup function runs once when you press reset or power the board. Thanks to this function, I initialize the digital pins (where the leds are plugged) as output and analog pin (where the potentiometer is plugged) as input. Now My Adruino knows the outputs and the input. **For the Last part of the code**, I used the loop() function. The loop function runs over and over again forever. With this function, I read my potbuttonval with the analogRead() function (she reads the value from the specified analog pin). And then I simply made my leds turning on and off according to the value of the button and with a 100 millisecond delay.

Link for ex 2: https://www.tinkercad.com/things/iPIbbbvkSyO-exercice-2/editel?sharecode=PB6fGP1NhCx5AJpobp5J5_IU-FdiYsfUuaxt4184Hr0

EX3

```
// Cyrille's CODE
//First Part
const int numbofLeds = 10; // total number of LEDs
const int ledPins[] = { 2, 3, 4, 5, 6, 7, 8, 9, 10, 11}; // like a set
of LEDs pins
const int pot=A0; //the pin where the potentiometer is plugged
const boolean LED_ON = LOW; // to turn the LED on in a more fluid way
const boolean LED_OFF = HIGH; // to turn the LED off in a more fluid
way (not one by one)
int buttonValue = 0; // value read from the button that will change
during the process by taking the value of the position of the
potentiometer's' Button. At the start of the program, its value should
be 0
int ledLevel = 0; // button value converted into LED sequence

//Second Part
void setup() {
  for (int led = 0; led < numbofLeds; led++)
  {
    pinMode(ledPins[led], OUTPUT); // make all the LED pins outputs by
using a for loop
    pinMode(pot, INPUT);// define the potentiometer's' pin as input
(because that's' the thing that we will change in order to modify the
brightness of the leds)
  }
}
```

```
//Last Part
void loop() {
  buttonValue = analogRead(pot); // read the value from the analog pin
where the potentiometer is plugged (here it's' A0)
  ledLevel = map(buttonValue, 0, 1023, 0, numbofLeds); // map to the
number of LEDs from one range to another
  for (int led = 0; led < numbofLeds; led++)
  {
    if (led < ledLevel ) {//the condidion
      digitalWrite(ledPins[led], LED_ON); // turn on pins less than
the level if the condition is met
    }
    else {
      digitalWrite(ledPins[led], LED_OFF); // turn off pins higher
than the level if the condition is not met
    }
  }
}//end
```

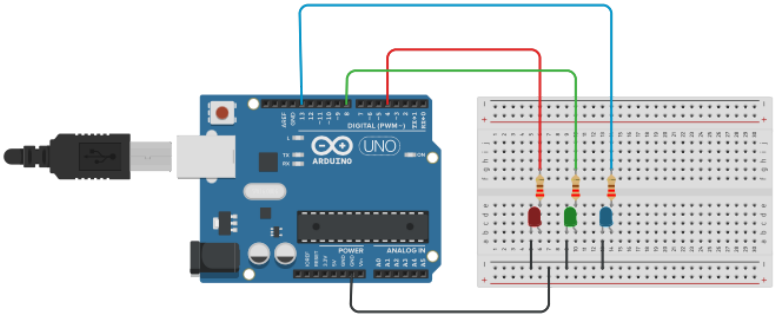
**SUMMARY**: I used Rajeev's explanations from the robotics class and the reference circuit to set up de device. I had 10 leds with 10 resistors of 220 ohm connected to the Adruino uno R3 with wires and a breadboard. I also added a 10 kilo ohm Potentiometer that will control the leds brightness. I chose to use different colors to make my work easier to understand. The leds were plugged on ramdomly chosen Digital pins and the Ground pin (GND). And the Potentiometer on an analog pin. The power chosen was 5V.

**For the first part of the code**, I just defined some variables so that I can tell my device where the LEDs and the Potentiometer are connected. I also decided to set a counter called 'buttonValue' and to convert it in led sequence. I didn't use integer for the led variable in order to make the variations more fluid. **For the Second part of the code**, I used the setup() function. The setup function runs once when you press reset or power the board. Thanks to this function, I initialize the digital pins (where the leds are plugged) as outputs and the analog pin as input by using a for loop. Now My Adruino knows the outputs and the input. **For the Last part of the code**, I used the loop() function. The loop function runs over and over again forever. With this function, I read my buttonvalue with the analogRead() function (she reads the value from the specified analog pin). I also used the map() function. The map function will calculate the number of LEDs that should be lit as a proportion of the button value. The code loops through each LED, turning it on if the proportional value of the button is greater than the LED number. For example, if the button value is 0, all the pins are off and when the button is at maximum value, all the LEDs will be lit. And then I simply made my leds turning on and off according to the value of the button and my conditions (which depends on the led sequence). The use of the for loop allows me to control the leds.

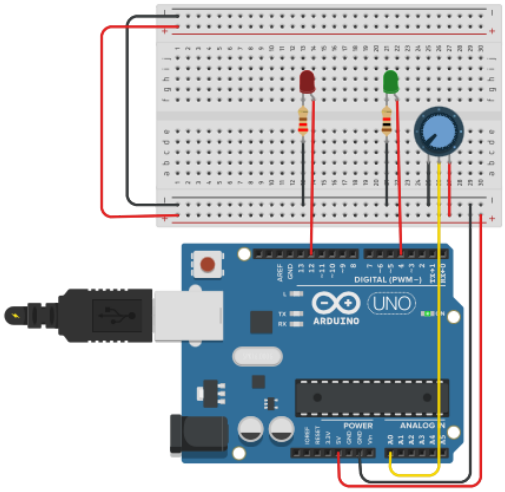
Links for ex 3 : https://www.tinkercad.com/things/570uu9iqqfK-exercice-3/editel?sharecode=qUYLyWYDiNB2cKCNoS2uNDQM4T93tYDwDKFwneZE_Gs

# Circuits and plots

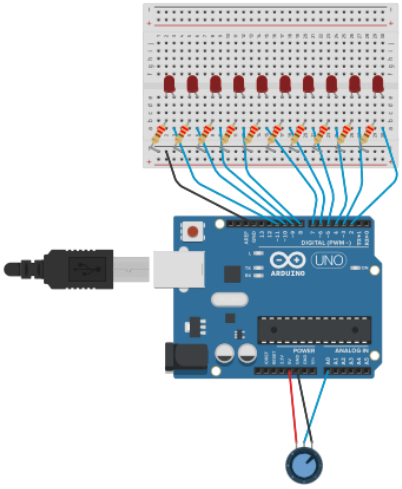
Ex1



Ex 2

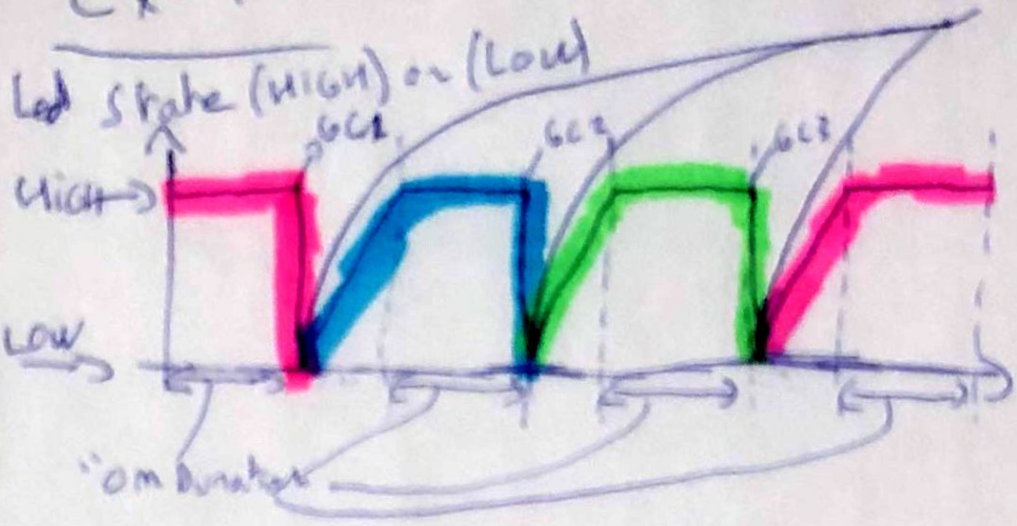


Ex3



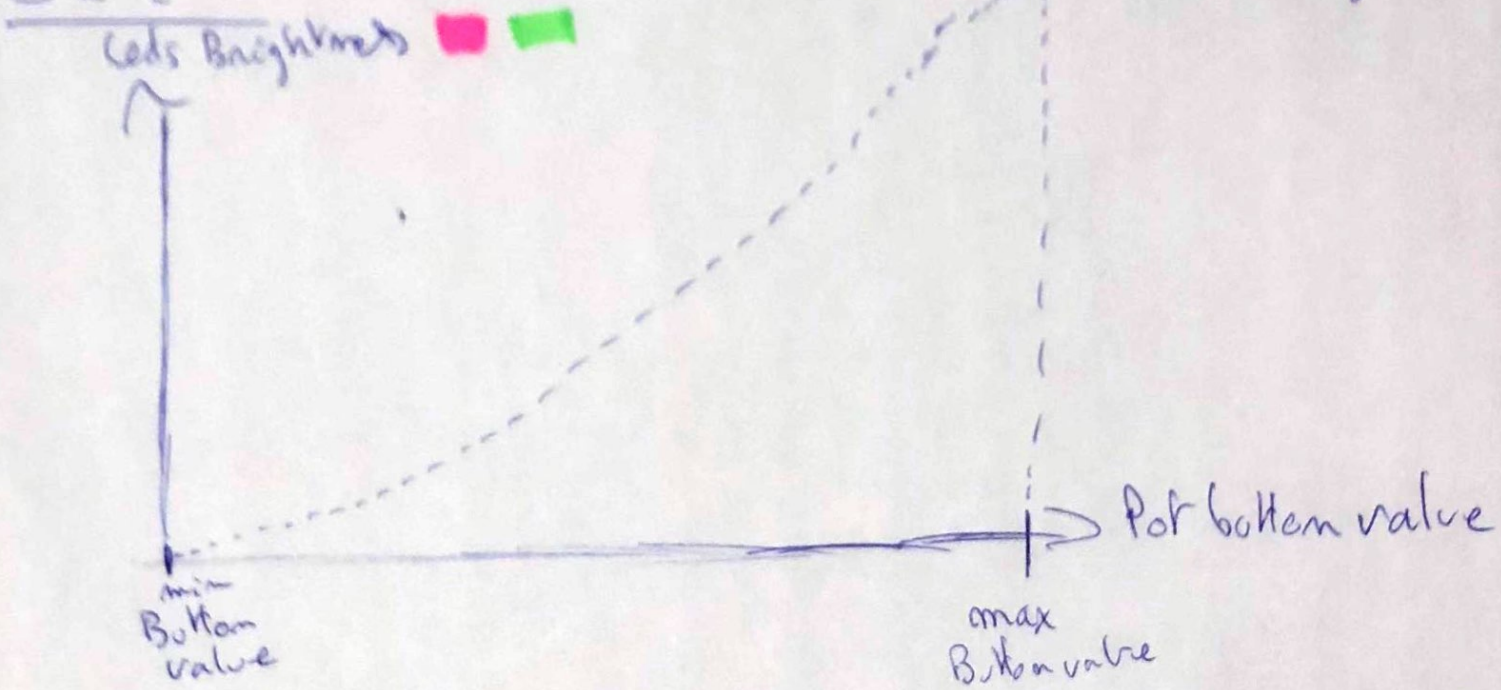
Ex 1:

"the moment when the specific conditions are met"




GC = "General condition"

Ex 2:




Ex3: Leds Brightness (x10)