Compte Rendu Examen

Sujet: 04

Partie I :

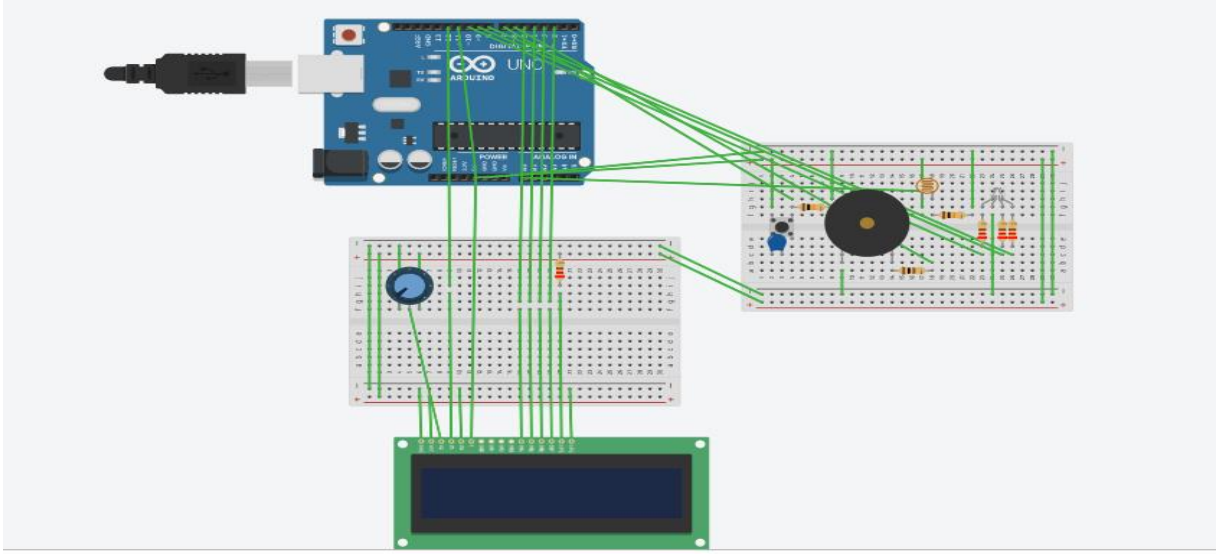
Une Led RGB qui s'allume en vert lorsque la luminosité dépasse 900 , en bleu lorsque celle est comprise entre 600 et 900 en rouge lorsque la luminosité est entre 350 et 600 et en blanc lorsque la luminosité est inférieure à 350.

Un Bouton poussoir qui allume la Led en blanc et actionne le buzzer pour une seconde. Un autre pousse pour actionner le buzzer et éteindre la Led RGB.

Hardware

- Arduino or Genuino Board
- Momentary button or Switch
- 1 Capacitor
- 1 Piezo
- 1 RGB LED
- 1 Potentiometer
- 1 LCD 16*2
- 2 10K ohm resistor
- 100 ohm resistor
- 4 220 ohm resistor
- hook-up wires
- 2 breadboards

Circuit



Code

```
#include <LiquidCrystal.h>

// on inclut la librairie

// initialise l'écran avec les bonnes broches

// ATTENTION, REMPLACER LES NOMBRES PAR VOS BRANCHEMENTS À VOUS !

LiquidCrystal Ecran(12,11,5,4,3,2);

const int buttonPin = 7; // the number of the pushbutton pin

const int ledRouge = 10; // the number of the RED LED pin

const int ledVert = 8; // the number of the GREEN LED pin
```

```
const int ledBleu = 9; // the number of the BLUE LED pin
const int buzzer =6; // Buzzer Pin
int buzzerState = HIGH; //Initial Buzzer State
int ledState = LOW;
int buttonState;
int lastButtonState = LOW;
unsigned long lastDebounceTime = 0; // the last time the output pin was toggled
unsigned long debounceDelay = 50; // the debounce time; increase if the output
void setup() {


  Serial.begin(9600);
  /**if(analogRead(A0) <350)
  {
    lcd.setCursor(0,1); //on passe à la ligne suivante
      lcd.print("Eclairage actif"); // on finit d'écrire
  }  **/
  pinMode(buzzer, OUTPUT);
```

```
  pinMode(buttonPin, INPUT);
  pinMode(ledRouge, OUTPUT);
  pinMode(ledVert, OUTPUT);
  pinMode(ledBleu, OUTPUT);
  // set initial LED state
  digitalWrite(ledRouge, ledState);
  digitalWrite(ledVert, ledState);
  digitalWrite(ledBleu, ledState);
  Ecran.begin(16,2);

}
void loop()
{
    int reading = digitalRead(buttonPin);
    if (reading != lastButtonState) {
    // reset the debouncing timer
    lastDebounceTime = millis();
}
  if ((millis() - lastDebounceTime) > debounceDelay) {
```

```
// whatever the reading is at, it's been there for longer
than the debounce
// delay, so take it as the actual current state:
// if the button state has changed:
if (reading != buttonState) {

    buttonState = reading;
    // only toggle the LED if the new button state is HIGH
    if (buttonState == HIGH) {

        ledState = !ledState;
          digitalWrite(ledRouge,ledState);
        digitalWrite(ledVert,ledState);
        digitalWrite(ledBleu,ledState);
        tone(buzzer, 1000);
        delay(1000);
        noTone(buzzer);
    }
  }
}
```

```
// save the reading. Next time through the loop, it'll be the lastButtonState:
lastButtonState = reading;
int valeur = analogRead(A0);
Ecran.setCursor(0, 0);
Ecran.print("Luminosite: ");
Ecran.print((int)valeur);
//if(ledState)
//{
if(valeur > 900)
{
  Ecran.setCursor(0, 1);
  Ecran.print("                ");
  Serial.println("il fait jour");
  Serial.println(valeur);
  digitalWrite(ledRouge,LOW);
  digitalWrite(ledVert,HIGH);
  digitalWrite(ledBleu,LOW);

}
```

```
  if(valeur > 600 && valeur < 900) {

   Ecran.setCursor(0, 1);
   Ecran.print("               ");
   Serial.println(valeur);
   digitalWrite(ledBleu,HIGH);
   digitalWrite(ledRouge,LOW);
   digitalWrite(ledVert,LOW);
}
  if(valeur > 350 && valeur < 600) {

   Ecran.setCursor(0, 1);
   Ecran.print("               ");
   Serial.println("il fait nuit");
   Serial.println(valeur);
   digitalWrite(ledRouge,HIGH);
   digitalWrite(ledVert,LOW);
   digitalWrite(ledBleu,LOW);

  }
```

```
  if(valeur < 350 && valeur >0) {
  Ecran.setCursor(0, 1);
  Ecran.print("Eclairage Actif");
  Serial.println("il fait nuit");
  Serial.println(valeur);
  digitalWrite(ledRouge,HIGH);
  digitalWrite(ledVert,HIGH);
  digitalWrite(ledBleu,HIGH);

 }
 delay(250);
//}
}
```

Partie II :

```
/*
This example shows how to connect to Cayenne using an Ethernet W5100 shield and send/receive sample data.
```

The CayenneMQTT Library is required to run this sketch. If you have not already done so you can install it from the Arduino IDE Library Manager.

Steps:

1. Set the Cayenne authentication info to match the authentication info from the Dashboard.

2. Compile and upload the sketch.

3. A temporary widget will be automatically generated in the Cayenne Dashboard. To make the widget permanent click the plus sign on the widget.

*/

//#define CAYENNE_DEBUG   // Uncomment to show debug messages

#define CAYENNE_PRINT Serial // Comment this out to disable prints and save space

#define SENSOR_PIN 0

#define VIRTUAL_CHANNEL1 1

#include <CayenneMQTTEthernet.h>

// Cayenne authentication info. This should be obtained from the Cayenne Dashboard.

```
char username[] = "973344d0-1691-11ea-a38a-d57172a4b4d4";

char password[] = "a74129630f79da884d6a6a52ced55856958aaa17";

char clientID[] = "c5175910-32e6-11ea-a38a-d57172a4b4d4";


#define VIRTUAL_CHANNEL2 2

#define LED_PIN 9 // Do not use digital pins 0 or 1 since those conflict with the use of Serial.

#define LED_PIN1 10 // Do not use digital pins 0 or 1 since those conflict with the use of Serial.

#define LED_PIN2 11 // Do not use digital pins 0 or 1 since those conflict with the use of Serial.


void setup() {
  Serial.begin(9600);
  Cayenne.begin(username, password, clientID);
}

void loop() {
  Cayenne.loop();
}
```

```
// Default function for sending sensor data at intervals to Cayenne.
// You can also use functions for specific channels, e.g CAYENNE_OUT(1) for sending channel 1 data.
CAYENNE_OUT_DEFAULT()
{
  // Write data to Cayenne here. This example just sends the current uptime in milliseconds on virtual channel 0.
  Cayenne.virtualWrite(0, millis());
  // Some examples of other functions you can use to send data.
  //Cayenne.celsiusWrite(1, 22.0);
  //Cayenne.luxWrite(2, 700);
  //Cayenne.virtualWrite(3, 50, TYPE_PROXIMITY, UNIT_CENTIMETER);
}

// Default function for processing actuator commands from the Cayenne Dashboard.
// You can also use functions for specific channels, e.g CAYENNE_IN(1) for channel 1 commands.
```

```
CAYENNE_IN_DEFAULT()
{
  CAYENNE_LOG("Channel %u, value %s", request.channel, getValue.asString());
  //Process message here. If there is an error set an error message using getValue.setError(), e.g getValue.setError("Error message");
}
// This function is called at intervals to send sensor data to Cayenne.
CAYENNE_OUT(VIRTUAL_CHANNEL1)
{
  Cayenne.virtualWrite(VIRTUAL_CHANNEL1, analogRead(SENSOR_PIN), "analog_sensor", "null");
}
// This function is called when data is sent from Cayenne.
CAYENNE_IN(VIRTUAL_CHANNEL2)
{
  int value = getValue.asInt();
  CAYENNE_LOG("Channel %d, pin %d, value %d", VIRTUAL_CHANNEL2, LED_PIN, value);
```

```
  // Write the value received to the digital pin.
  digitalWrite(LED_PIN, value);

}
```