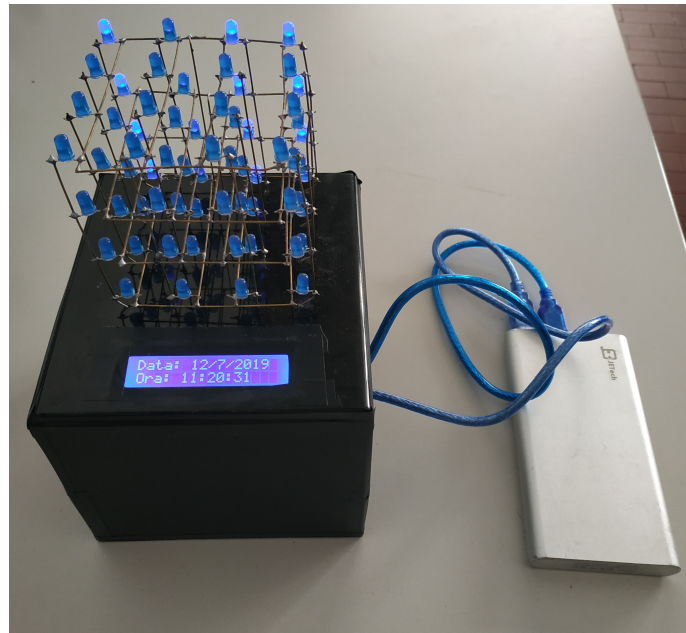


Progetto ARE: Cubo di LED con Orologio integrato con modulo RTC



Introduzione

L'idea di base era di creare un cubo di LED con diversi pattern in modo tale da realizzare diverse coreografie; a questo abbiamo deciso di aggiungere un display LCD che segnasse la data e l'ora tramite un modulo RTC (Real Time Clock).

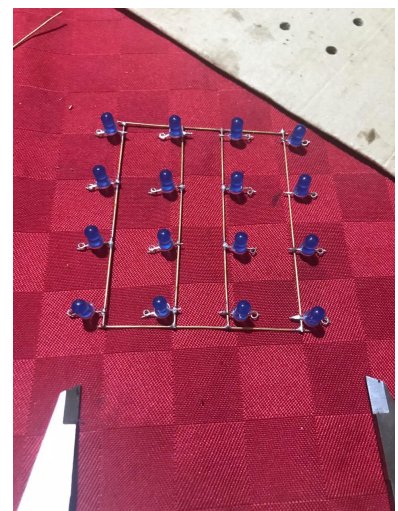
Materiali

- N° 2 modelli Arduino "UNO";
- N° 64 LED di colore blu;
- Filo di ottone;
- N° 4 resistenze da 100 Ohm;
- N° 1 LCD 2x16;
- N° 1 potenziometro a rotazione, 10k Ohm;
- N° 1 resistenza da 220 Ohm;
- N° 1 modulo RTC;
- Varia cavetteria;
- N° 1 box in plastica per contenere il tutto;
- N° 1 Powerbank per alimentare gli Arduino;

Sviluppo

È stata creata una struttura a griglia per contenere tutti i LED di ogni piano, collegando alla suddetta griglia i catodi di ogni LED. Gli anodi invece sono stati collegati insieme colonna per colonna. Una volta creato il cubo, tramite delle colonnine di filo di rame sono stati effettuati i collegamenti per l'alimentazione dei LED: queste colonnine sono state collegate una in ogni piano e all'altra estremità sono state saldate le resistenze da 100 Ohm.

Per quanto riguarda i collegamenti, sono stati utilizzati quasi tutti i pin di arduino, in modo che ogni LED ricevesse il segnale in combinazione dalla colonna e dal piano.



```

void setup()
{
  for(int i = 0; i<16; i++)
  {
    pinMode(column[i], OUTPUT); //setting rows to output
  }

  for(int i = 0; i<4; i++)
  {
    pinMode(layer[i], OUTPUT); //setting layers to output
  }

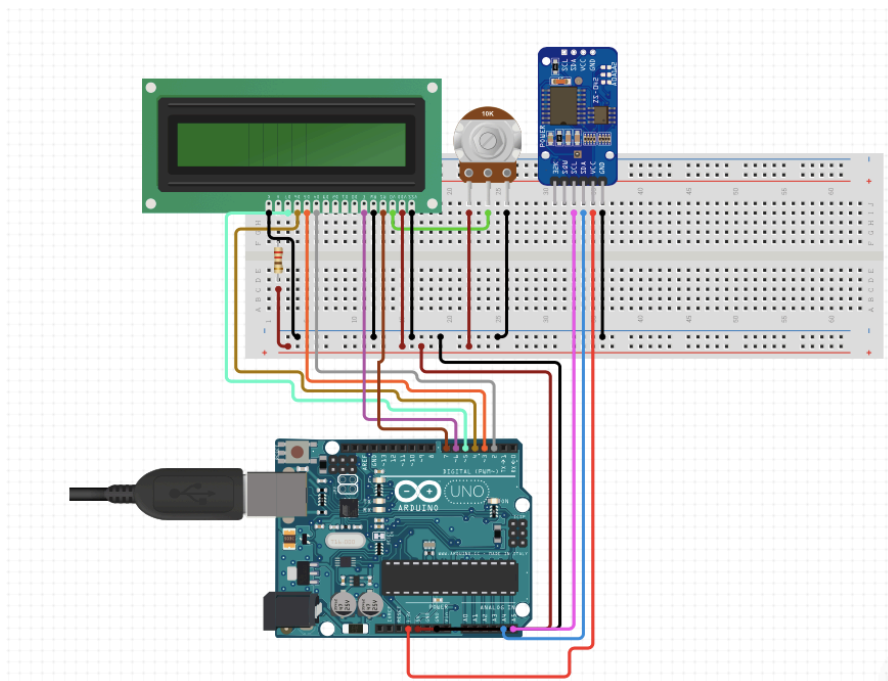
  randomSeed(analogRead(10)); //seeding random for random pattern
}

void loop()
{
  turnEverythingOff();
  flickerOn();
  turnEverythingOn();
  delay(time);
  turnOnAndOffAllByLayerUpAndDownNotTimed();
  layerstompUpAndDown();
  spiralInAndOut();
  turnOnAndOffAllByColumnSideways();
  delay(time);
  aroundEdgeDown();
  turnEverythingOff();
  randomflicker();
  randomRain();
  diagonalRectangle();
  goThroughAllLedsOneAtATime();
  propeller();
  spiralInAndOut();
  flickerOff();
  turnEverythingOff();
  delay(2000);
}

```

A livello di codice, abbiamo creato vari pattern che si ripetono in loop.

Per quanto riguarda l'orologio, al secondo Arduino abbiamo collegato il display LCD che necessita in un potenziometro per dare la giusta luminosità al display. Il modulo RTC è dotato di una pila che permette di continuare a lavorare, e quindi segnare l'ora esatta, anche quando l'Arduino non riceve direttamente l'alimentazione. Infatti nel video si può vedere che staccando l'alimentazione e riattaccandola, il display segna l'ora esatta e non si ferma.



```

void setup() {
  Serial.begin(9600);
  while (!Serial);
  Serial.println("start");
  lcd.begin(16, 2);
  if (! rtcDS.begin()) {
    Serial.println("Couldn't find RTC");
    while (1);
  }
  if (rtcDS.lostPower()) {
    Serial.println("RTC lost power, lets set the time!");
    //La seguente linea inizializza data e ora
    rtcDS.adjust(DateTime(F(__DATE__), F(__TIME__)));
  }
}

void loop() {
  //Questo mostrerà la data e l'ora nel display
  DateTime now = rtcDS.now();
  lcd.setCursor(0,0);
  lcd.print("Data: ");
  lcd.print(now.day(), DEC);
  lcd.print('/');
  lcd.print(now.month(), DEC);
  lcd.print('/');
  lcd.print(now.year(), DEC);
  lcd.print(" ");
  lcd.setCursor(0,1);
  lcd.print("Ora: ");
  lcd.print(now.hour(), DEC);
  lcd.print(':');
  lcd.print(now.minute(), DEC);
  lcd.print(':');
  lcd.print(now.second(), DEC);

  delay(1000);
  lcd.clear();
}

```

Anche qui a livello di codice, utilizzando le opportune librerie per il modulo RTC, abbiamo fatto in modo che prendesse l'ora e la data esatta direttamente dal computer che iniettava il codice nell'Arduino.

Per quanto riguarda il refresh dell'immagine, è tutto gestito nel loop().

Nel video allegato si può vedere il progetto in funzione.