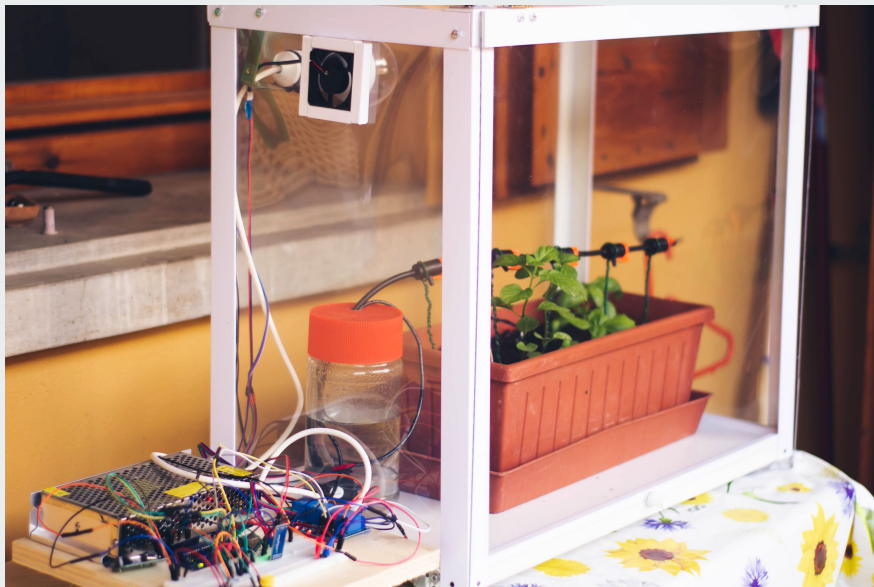


# Serra\_OS



Controllo e gestione  
dei parametri ambientali di una serra  
con Arduino

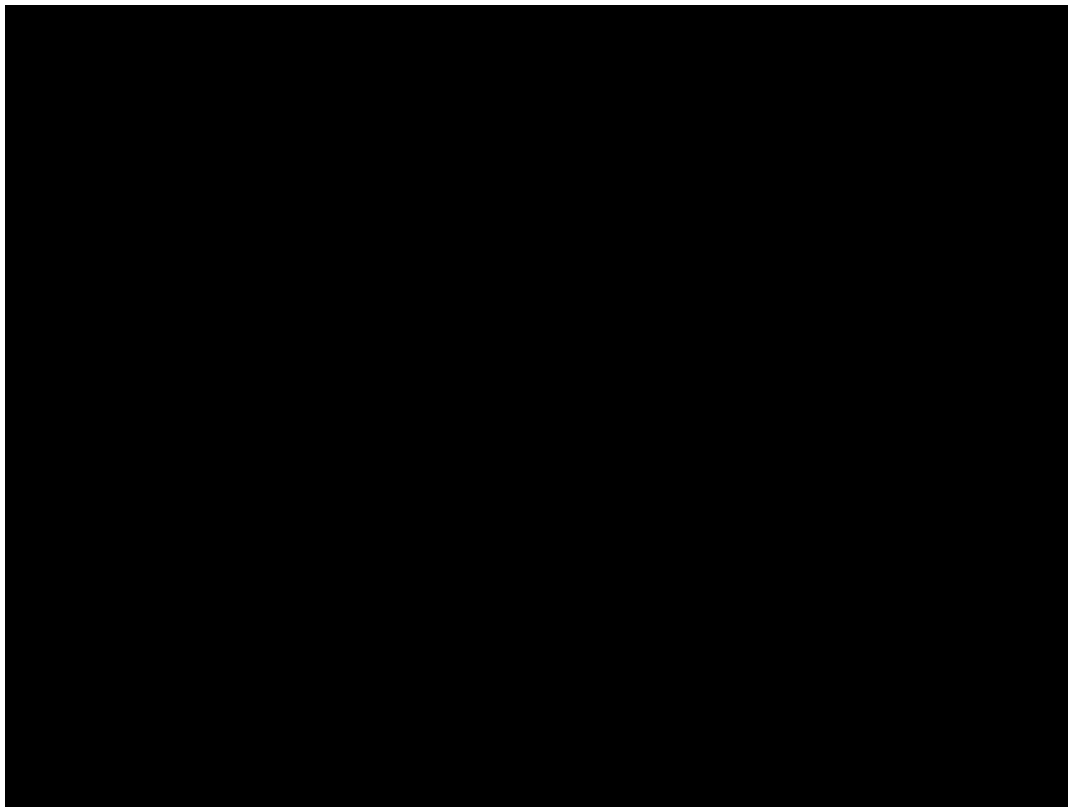
Gabriele Basigli

---

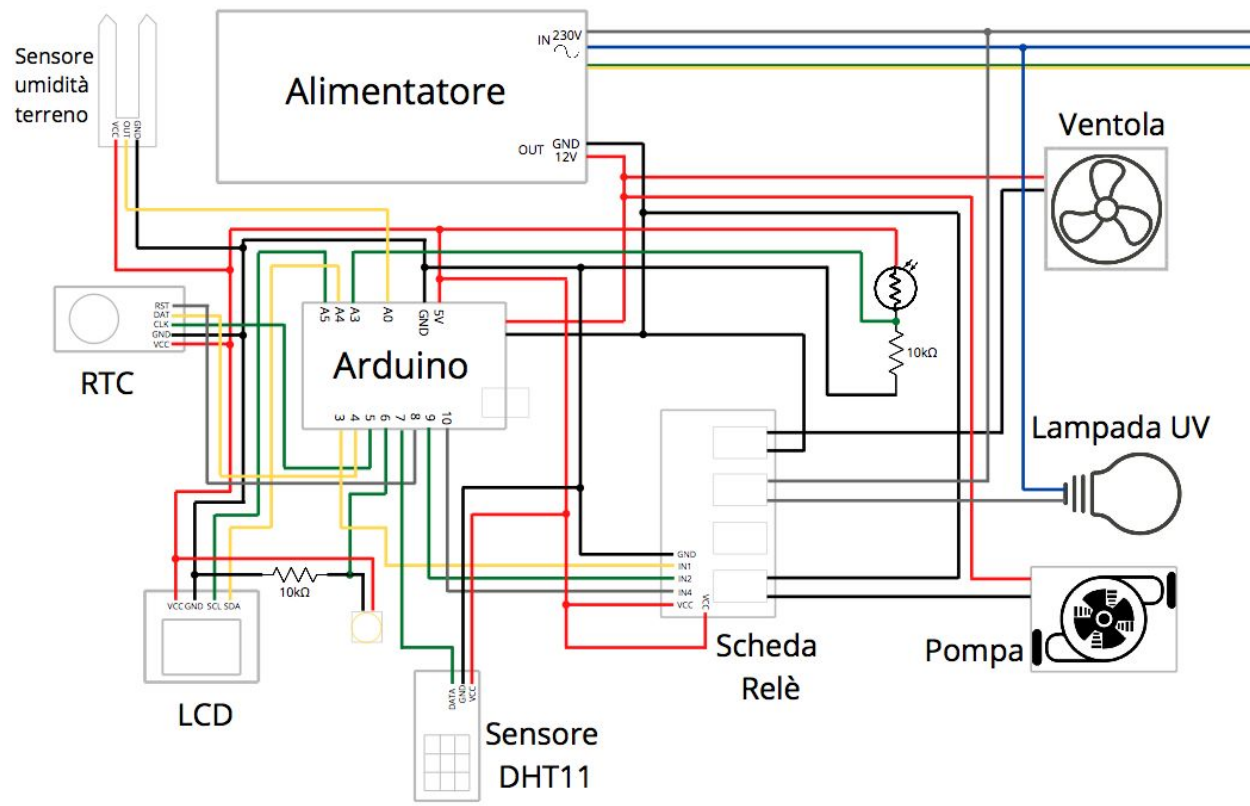
# SOMMARIO

- Automa gestione del display e chiamata funzioni
- Display interfaccia generale
- Display interfacce nel dettaglio
- Immagini a display
- Funzione controllo illuminazione
- Funzione controllo temperatura e umidità
- Funzione controllo irrigazione
- Il futuro del progetto

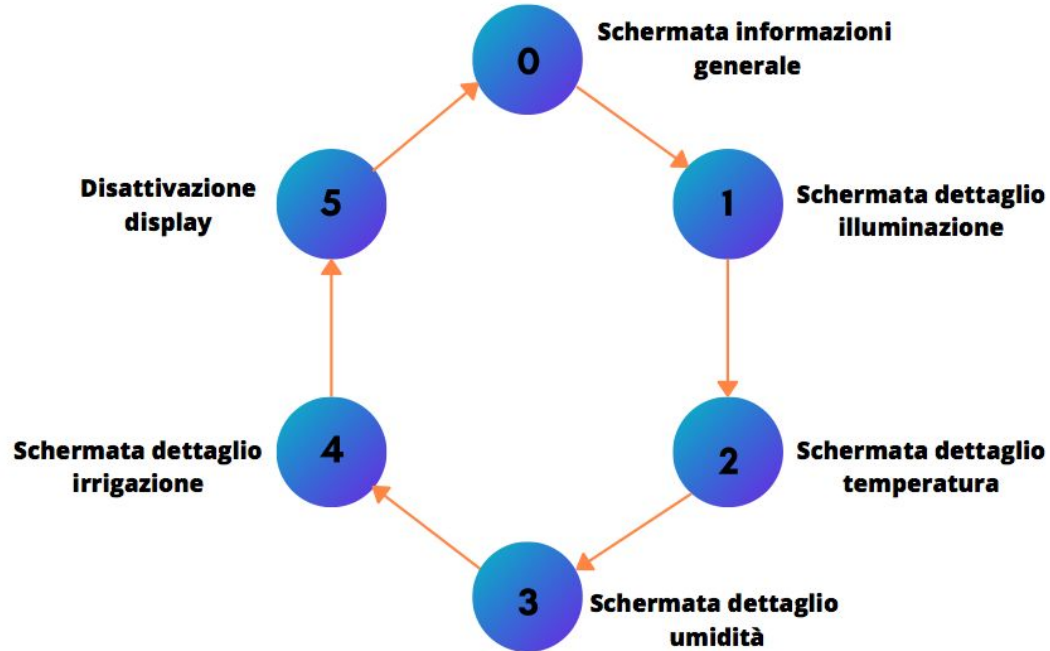
# Video introduttivo



# Schema cablaggio



# Automa gestione del display e chiamata funzioni



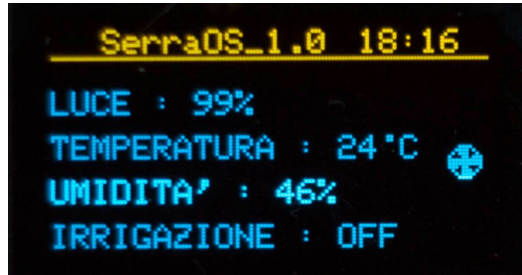
```
void OS(){
    dt3 = millis() - t3;
    if(dt3 >= 50){
        t3 = millis();
        tasto = digitalRead(pinTasto); //fronte salita
        if(!prev_tasto && tasto){
            stato = stato + 1;
            if (stato > 5){
                stato = 0;
            }
            Serial.println(stato);
            prev_tasto = tasto;
        }
        if(prev_tasto && !tasto){
            prev_tasto = tasto;
        }
    }

    dt1 = millis() - t1;
    if (dt1 >= 1000){
        t1 = millis();
        Display();
    }

    illuminazione();
    TempHum();
    Irrigazione();
}
```

# Display interfaccia generale

- Rapida lettura di tutti i valori



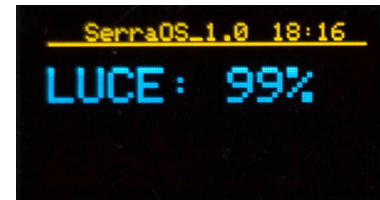
- Gestione Real Time Clock

```
void Display(){
    switch (stato){
        case 0:
            //-----INTERFACCIA GENERALE-----
            //-----ILLUMINAZIONE-----
            display.clearDisplay();
            display.setTextSize(0);
            display.setCursor(15, 0);
            display.println("SerraOS_1.0");
            display.setCursor(90, 0);
            RTC.updateTime();
            display.print(RTC.hours);
            display.print(":");
            display.print(RTC.minutes);
            display.drawLine(0, 8, 200, 8, WHITE); // Draw line (x0,y0,x1,y1,color)
            display.setCursor(0, 16);
            display.print("LUCE : ");
            display.print(Lperc);
            display.print("%");
            if (Vluci == 1){
                display.drawBitmap(115, 15, logo12_lamp_bmp, 12, 12, 1);
            }
            if (Vvent == 1){
                display.drawBitmap(115, 30, fan12_lamp_bmp, 12, 12, 1);
            }
            //-----TEMPERATURA E UMIDITA'-----
            display.setCursor(0, 29);
            display.print("TEMPERATURA : ");
            display.print(TEMP);
            display.drawRect(98, 29, 2, 2, WHITE); // Draw rectangle (x,y,width,height,color)
            display.print(" °C");
            display.setCursor(0, 42);
            display.print("UMIDITA' : ");
            display.print(HUM);
            display.print("%");
            //-----IRRIGAZIONE-----
            display.setCursor(0, 55);
            display.print("IRRIGAZIONE : ");
            if (Vpump == 1){
                display.print("ON");
            }
            if (Vpump == 0){
                display.print("OFF");
            }
            display.display();
            break;
    }
}
```

# Display dettaglio illuminazione

```
case 1:          //dettaglio luce
oneTime4 = true;
if (oneTime1){
    display.clearDisplay();
    display.setTextSize(0);
    display.drawBitmap(0, 0, logo_biglamp_bmp, 128, 64, 1);
    display.display();
    delay(1000);
    oneTime1 = false; ←
}
display.clearDisplay();
display.setTextSize(0);
display.setCursor(15, 0);
display.println("SerraOS_1.0");
display.setCursor(90, 0);
RTC.updateTime();
display.print(RTC.hours);
display.print(":");
display.print(RTC.minutes);
display.drawLine(0, 8, 200, 8, WHITE); // Draw line (x0,y0,x1,y1,color)
display.setTextSize(2);
display.setCursor(0, 16);
display.print("LUCE: ");
display.print(Lperc);
display.print("%");
if (VLuci == 1){
    display.setTextSize(0);
    display.setCursor(5, 40);
    display.print("Illuminazione UV ON");
    display.drawBitmap(54, 50, logo12_lamp_bmp, 12, 12, 1);
}
display.display();
break;
```

- Variabile oneTime
- Dettaglio stato illuminazione e accensione lampada UV



# Display dettaglio temperatura

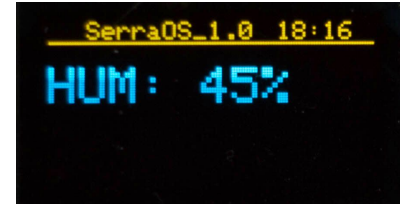
```
case 2: //dettaglio temperatura
    oneTime1 = true;
    if (oneTime2){
        display.clearDisplay();
        display.setTextSize(0);
        display.drawBitmap(0, 0, TEMPlogo_bmp, 128, 64, 1);
        display.display();
        delay(1000);
        oneTime2 = false;
    }
    display.clearDisplay();
    display.setTextSize(0);
    display.setCursor(15, 0);
    display.println("SerraOS_1.0");
    display.setCursor(90, 0);
    RTC.updateTime();
    display.print(RTC.hours);
    display.print(":");
    display.print(RTC.minutes);
    display.drawLine(0, 8, 200, 8, WHITE); // Draw line (x0,y0,x1,y1,color)
    display.setTextSize(2);
    display.setCursor(0, 16);
    display.print("TEMP: ");
    display.print(TEMP);
    display.drawRect(100, 16, 4, 4, WHITE); // Draw rectangle (x,y,width,height,color)
    display.print(" °C");
    display.display();
    break;
```





# Display dettaglio umidità

```
case 3:    //dettaglio umidità
  oneTime2 = true;
  if (oneTime3){
    display.clearDisplay();
    display.setTextSize(0);
    display.drawBitmap(0, 2, HUMlogo_bmp, 128, 64, 1);
    display.display();
    delay(1000);
    oneTime3 = false;
  }
  display.clearDisplay();
  display.setTextSize(0);
  display.setCursor(15, 0);
  display.println("SerraOS_1.0");
  display.setCursor(90, 0);
  RTC.updateTime();
  display.print(RTC.hours);
  display.print(":");
  display.print(RTC.minutes);
  display.drawLine(0, 8, 200, 8, WHITE); // Draw line (x0,y0,x1,y1,color)
  display.setTextSize(2);
  display.setCursor(0, 16);
  display.print("HUM: ");
  display.print(HUM);
  display.print("%");
  display.display();
break;
```



# Display dettaglio irrigazione

- Valore di umidità del terreno



```
case 4: //dettaglio irrigazione
    oneTime3 = true;
    if (oneTime4){
        display.clearDisplay();
        display.setTextSize(0);
        display.drawBitmap(0, 6, IRRlogo_bmp, 128, 64, 1);
        display.display();
        delay(1000);
        oneTime4 = false;
    }
    display.clearDisplay();
    display.setTextSize(0);
    display.setCursor(15, 0);
    display.println("SerraOS_1.0");
    display.setCursor(90, 0);
    RTC.updateTime();
    display.print(RTC.hours);
    display.print(":");
    display.print(RTC.minutes);
    display.drawLine(0, 8, 200, 8, WHITE); // Draw line (x0,y0,x1,y1,color)
    display.setTextSize(2);
    display.setCursor(0, 16);
    display.println("HUM ");
    display.print("TERRA: ");
    display.print(ValoreTerreno);
    //display.print(TerrenoPerc);
    //display.println("%");
    display.setTextSize(0);
    display.setCursor(0, 55);
    display.print("IRRIGAZIONE: ");
    if (Vpump == 1){
        display.print("ON");
    }
    if (Vpump == 0){
        display.print("OFF");
    }
    display.display();
    break;
case 5:
    display.clearDisplay();
    display.display();
}
```

# Immagini a display

## Bitmap Converter

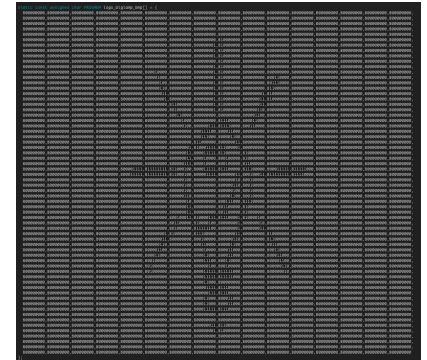
Convert image files and data into Marlin Boot Screens and Status Screen Logos for Graphical Displays. Dark colors show up as "on" pixels. Light colors will be transparent.



Conversione in  
binario



Disegno con  
Photoshop



static const unsigned char PROGMEM

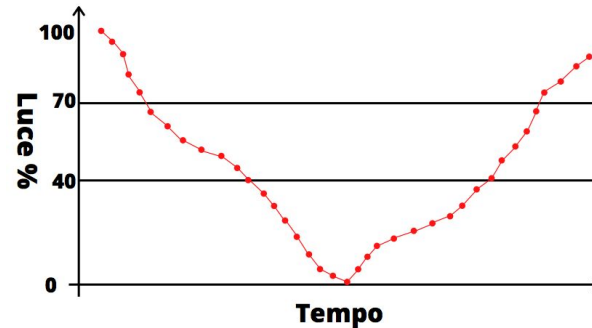
# Funzione controllo illuminazione



```
void illuminazione(){
  dt2 = millis() - t2;
  if (dt2 >= 1000){
    t2 = millis();
    for(int i = 0; i < 10; i++){
      L += analogRead(Iluci);
    }
    L = L/10;
    Lperc = map(L, 0, 1024, 0, 100);
  }
  if (Lperc < 40){
    Vluci = 1;
  }
  if (Lperc > 70){
    Vluci = 0;
  }
  digitalWrite(0luci, Vluci);
}
```

- Conversione in luce percentuale

- Isteresi



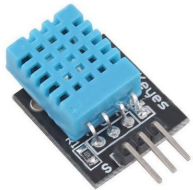
# Funzione controllo temperatura e umidità

☁☁☁ H%

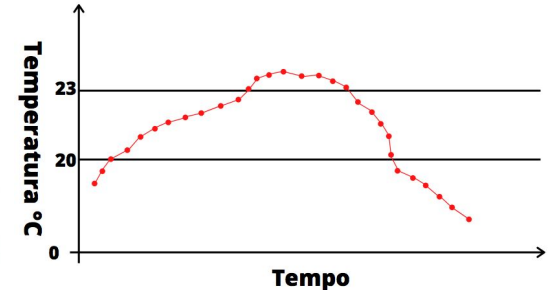
🌡 °C

```
void TempHum(){
  if ((err = dht11.read(PinTH, &TEMP, &HUM, NULL)) != SimpleDHTErrSuccess){
    return;
  }
  if (TEMP > 23){
    Vvent = 1;
  }
  if (TEMP < 20) {
    Vvent = 0;
  }
  digitalWrite(PinVentola, Vvent);
}
```

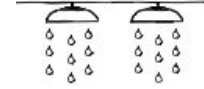
- Lettura dei valori di temperatura e umidità
- Isteresi controllo ON-OFF



- Sensore DHT11

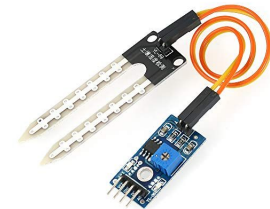


# Funzione controllo irrigazione



```
void Irrigazione(){
  ValoreTerreno = analogRead(pinTerreno);
  //Serial.println(ValoreTerreno);
  TerrenoPerc = map(ValoreTerreno, 950, 350, 0, 100);
  if (ValoreTerreno < 700){
    Vpump = 1;
  }
  if (ValoreTerreno > 800){
    Vpump = 0;
  }
  if (RTC.hours == 7 && RTC.minutes == 30){
    Vpump = 1;
  }
  if (RTC.hours == 7 && RTC.minutes == 33){
    Vpump = 0;
  }
  digitalWrite(PinPompa, Vpump);
}
```

- Lettura del valore di umidità del terreno
- Conversione in percentuale
- Gestione della pompa
- Problemi dovuti al sensore



# Il futuro del progetto...

- App per monitorare la serra e per cambiarne le impostazioni
- Alimentazione solare
- Sistema di riscaldamento
- Sensore di livello per il serbatoio

