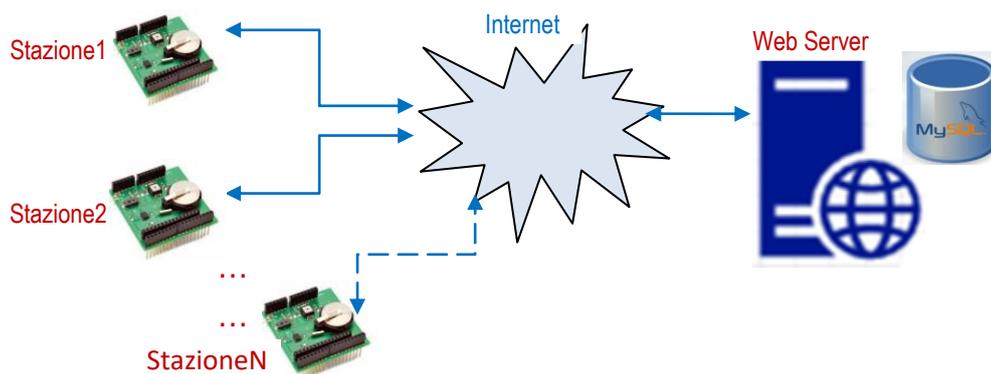


## Rete di stazioni Arduino che inviano dati ad un DB MySql remoto

Esempio applicativo di Internet of Things (IoT).

Il progetto realizza una rete di due stazioni di monitoraggio, basate su Arduino e collegate ad Internet, che rilevano la luminosità e inviano i valori letti ad un database MySql remoto. Sul server remoto, una pagina web presenta i dati memorizzati nel database sotto forma di tabelle.

Il progetto è volutamente semplice (le stazioni sono solo due e misurano unicamente la luminosità) e non è importante il tipo o il numero dei sensori, o la complessità dell'interfacciamento con Arduino; l'obiettivo è quello illustrare il funzionamento generale di un sistema di monitoraggio web dei dati, di qualsiasi natura e tipo di processo, provenienti da stazioni remote di rilevamento.



- Ciascuna stazione è basata su una scheda Arduino, interfacciata con una fotoresistenza, equipaggiata con Ethernet Shield e connessa a Internet. Ogni 10 minuti rileva il valore della luminosità e lo invia ad un database MySql remoto (applicazione Web Client Arduino '*stazione.ino*')
- Uno script PHP lato server riceve i valori inviati da Arduino e li inserisce nel database ( '*salvaluminosita.php*' )
- Una pagina PHP visualizza i dati memorizzati nel database ( '*displaydati.php*' )

Per la realizzazione e il test del progetto ho utilizzato:

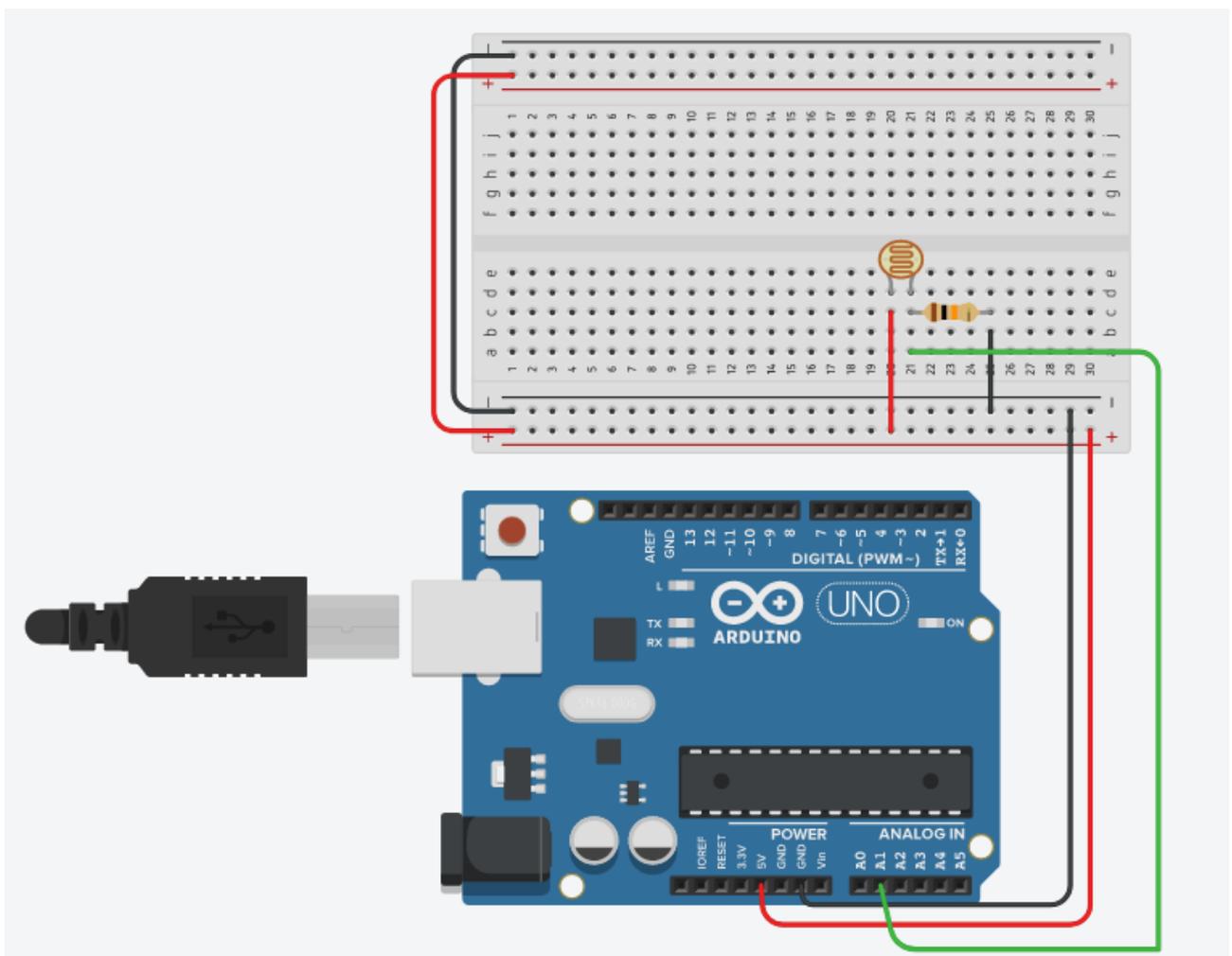
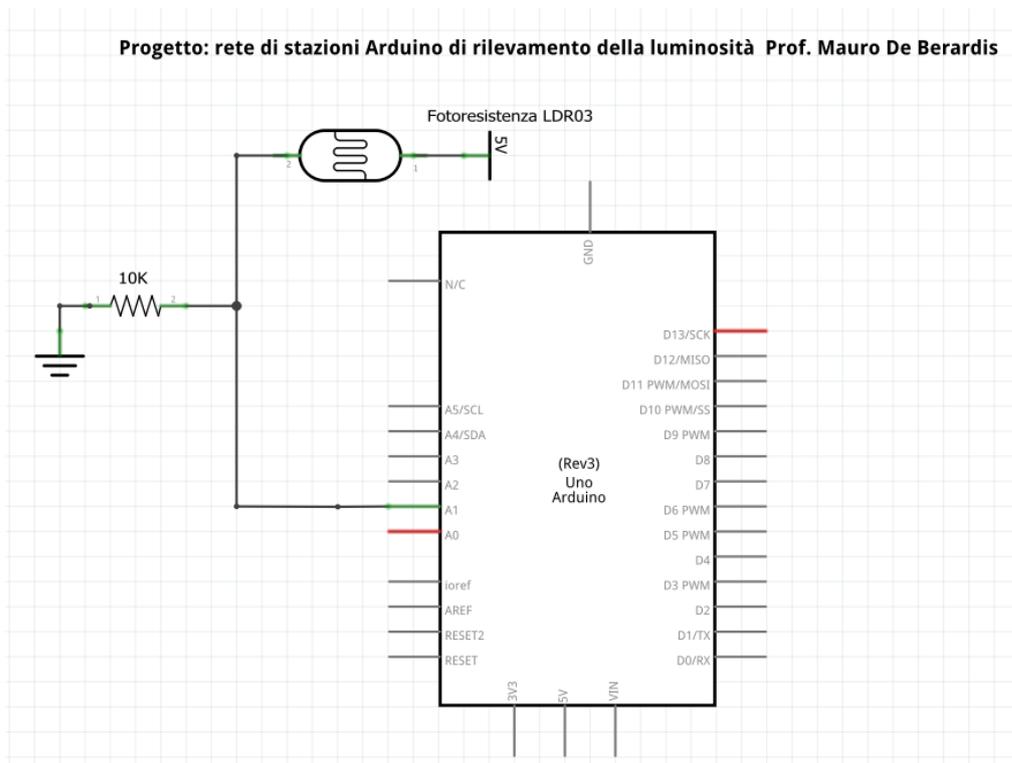
- un database MySql sul Server Linux che ospita il mio sito web [www.maurodeberardis.it](http://www.maurodeberardis.it)
- lo script [www.maurodeberardis.it/CodiciPHP/salvaluminosita.php](http://www.maurodeberardis.it/CodiciPHP/salvaluminosita.php)
- la pagina web [www.maurodeberardis.it/CodiciPHP/displaydati.php](http://www.maurodeberardis.it/CodiciPHP/displaydati.php)

**In alternativa ad un Web Server remoto si può utilizzare un Server locale.** Per questo motivo ho testato il progetto anche con un server locale WAMP e nei codici di programmazione, che fanno riferimento al test con Web Server remoto, ho inserito i commenti (da leggere attentamente!!) che spiegano in dettaglio come utilizzare WAMP.

Facendo riferimento anche ai materiali proposti nelle sezioni "Esercizi PHP", "HTML/CSS/JavaScript", "Arduino" e "App Android" dell'area download del sito [www.maurodeberardis.it](http://www.maurodeberardis.it), invito gli studenti più motivati a modificare il progetto:

- migliorando e integrando l'applicazione PHP che permette di visualizzare i dati rilevati dalle varie stazioni di monitoraggio
- realizzando un app Android che permetta ai dispositivi mobili di accedere al database remoto e visualizzare i dati di monitoraggio
- realizzando stazioni di monitoraggio più complesse, ad esempio stazioni meteo con sensori di temperatura, umidità, pressione atmosferica, pioggia, vento, direzione del vento ecc.

## Schema elettrico e collegamenti su breadboard di ciascuna stazione



## Componenti necessari

- 2 schede Arduino collegate via USB ad un PC Windows X con Arduino IDE
- 2 Ethernet Shield con "Ethernet Library"
- 2 Breadboard con cavetteria
- 2 Fotoresistenze
- 2 Resistenze da 10 KOhm

Per misurare la luminosità si utilizza una fotoresistenza, un dispositivo elettronico la cui resistenza varia in maniera inversamente proporzionale alla quantità di luce che lo colpisce.

Al buio il valore della resistenza è di qualche MOhm, in piena luce il valore scende a qualche KOhm.

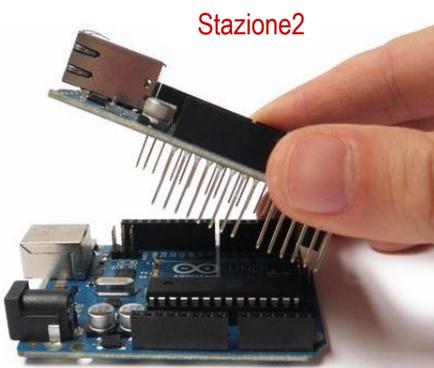
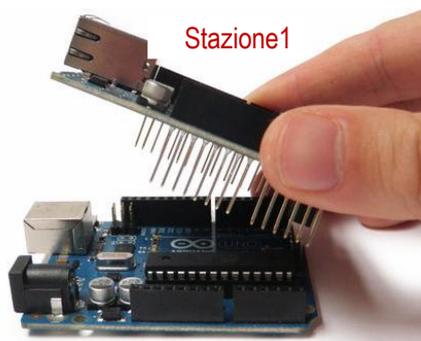
Per la misura utilizziamo un partitore di tensione, con la fotoresistenza e una resistenza da 10 kOhm, collegato all'ingresso analogico A1.

Quando Arduino legge con AnalogRead l'ingresso analogico A1, lo converte in digitale tramite un ADC (Convertitore analogico digitale) con risoluzione di 10 bit. I livelli di quantizzazione che l'ADC può fornire, sono  $2^{10} = 1024$ , ovvero sono tutte le combinazioni di 10 bit comprese tra 0000000000 e 1111111111 (in decimale da 0 a 1023).

Per semplicità, non ci preoccupiamo di convertire i valori forniti dall'ADC di Arduino in lumen o lux e assumiamo direttamente i valori letti con AnalogRead, tenendo conto che tali valori tendono a 1023 nel caso di luce piena e prolungata e a 0 nel caso di buio totale e prolungato.

## Circuito di prova

Dopo aver inserito Ethernet Shield e caricato lo sketch in entrambe le schede Arduino



utilizzando i piedini A1, 5V e Gnd di Arduino "replicati" su Ethernet Shield, realizziamo, per ciascuna stazione, i collegamenti previsti per realizzare il misuratore luminosità.

Collegiamo quindi le due Ethernet Shield allo switch di rete.

## Sketch di Arduino (Web Client)

Attenzione: **per ogni stazione si utilizza una scheda arduino e uno sketch**. Gli sketch sono uguali tranne che per i valori di configurazione: macaddress e ipstatico. Possono avere lo stesso nome ma anche, se si preferisce, nomi diversi (stazione1.ino, stazione2.ino): importante che siano diversi macaddress e ipaddress.

### Codice "stazione.ino"

```

/*Misuratore luminosità con Arduino ed Ethernet Shield - Mauro De Berardis 2021
SPI.h, Ethernet.h sono due librerie necessarie per realizzare il progetto.
Con Ethernet.h è possibile far diventare Arduino un mini server web
*/
#include <SPI.h>
#include <Ethernet.h>
int st=1;
/* int st=2; per la stazione n. 2 e così via*/

/* Le Ethernet shields vengono spesso fornite con un'etichetta che indica il Mac
address da utilizzare.
Le Ethernet shields che non hanno un Mac address preassegnato possono
essere configurate con un indirizzo di 6 cifre esadecimali qualsiasi, purchè diverso
dai MAC Address di altri dispositivi della LAN.
*/

byte macaddress[] = {0x80, 0x80, 0x80, 0xA80, 0x80, 0x80 };
// byte macaddress[] = {0xA0, 0x80, 0x80, 0xA80, 0x80, 0x80 };

/* Avendo a disposizione due Ethernet shields che non hanno un Mac address
preassegnato, ho scelto io i due Mac address: 80-80-80-80-80-80 e A0-80-80-80-80-80*/

char server[] = "www.maurodeberardis.it"; //Host del Server remoto

/* Se al posto del Web Server remoto si utilizza un server locale, ad esempio WAMP,
occorre indicare l'ip del pc in cui è installato il Server e occorre scrivere, ad
esempio: char server[] = "192.168.1.3";
*/

EthernetClient client;// istanziamo la classe EthernetClient nell'oggetto client

byte ipstatico[]={192,168,1,101};
/* eventuale indirizzo IP statico da utilizzare nel caso di fallimento della
connessione attraverso DHCP */
/* 192.168.1.101 è l'IP privato della scheda Arduino della stazione1
* 192.168.1.102 è l'IP privato della scheda Arduino della stazione2 ... e così via
*/
void setup()
{
  Serial.begin(115200); // impostare lo stesso valore sul monitor seriale

  if (Ethernet.begin(macaddress) == 0)
  {
    // se la connessione attraverso DHCP fallisce, utilizzo l'ip statico
    Serial.println("La connessione fallisce usando DHCP");
    Ethernet.begin(macaddress, ipstatico);
  }
  delay(1000);
}

```

```

void loop()
{
  /* Lettura luminosità: Arduino legge il valore della tensione del partitore
   * Fotoresistenza-10K, convertito in digitale dall'ADC a 10 bit
   */
  int lum=analogRead(A1);

  /* Una volta inizializzato il controller Ethernet, possiamo utilizzare la classe
   EthernetClient per poter comunicare con il server attraverso un socket.
   Ci connettiamo attraverso il metodo connect(URL,port): nel nostro caso l'URL è
   "www.maurodeberardis.it" e la porta è la 80 (porta di default).
   Se utilizziamo un Server locale, l'URL è l'indirizzo Ip locale (io ho
   utilizzato 192.168.1.3)
   */
  if (client.connect(server, 80))
  {
    * Se la connessione è ok, possiamo inviare la richiesta HTTP al server
    attraverso il metodo print(); il protocollo HTTP prevede l'invio di una
    richiesta GET o POST con una serie di headers, tra i quali "Host" per
    indicare l'host di destinazione (www.maurodeberardis.it o 192.168.1.3)
    */

    /*Scivo sul monitor seriale per sapere se la connessione è ok e visualizzare i
    valori che saranno memorizzati nel database MySql
    */
    Serial.println("Connessione OK");
    Serial.print("Stazione: ");
    Serial.println(st);
    Serial.print("Luminosità: ");
    Serial.println(lum);
    Serial.println();

    /*Richiesta GET e invio al database del numero di stazione e della luminosità
    rilevata. Nel caso di Web server remoto utilizzo lo script:
    www.maurodeberardis.it/CodiciPHP/salvaluminosita.php", Con Wamp Server
    utilizzo lo script 192.168.1.3/CodiciPHP/salvaluminosita.php
    */
    client.print("GET /CodiciPHP/salvaluminosita.php?");
    client.print("stazione=");
    client.print(st);
    client.print("&luminosita=");
    client.print(lum);
    client.println( " HTTP/1.1");
    client.print( "Host: " );
    client.println(server);
    client.println( "Connection: close" );
    client.println();
    client.println();
    client.println();
    client.stop();
  }
  else
  {
    Serial.println("Connessione fallita!");
  }
  delay(600000);
  /* attesa di 10 minuti ovvero di 600 secondi (600000 millisecondi) prima di inviare
  un'altra misura */
}

```

## Script PHP

Lo script php lato server riceve i valori inviati da Arduino lato client ed esegue una query di inserimento nella *tabella "misure"* del *database "stazioni"*: il database viene utilizzato ovviamente per memorizzare le letture del sensore, in modo che sia possibile accedervi in un secondo momento. È un database elementare con una sola tabella che memorizza un id, il numero della stazione che trasmette, la data e l'ora e il valore di luminosità.

### Query di creazione del database MySql "stazioni" e della tabella "misure"

```
CREATE DATABASE stazioni;
CREATE TABLE IF NOT EXISTS misure (
id INT(10) PRIMARY KEY AUTO_INCREMENT,
stazione INT(3),
dataora datetime,
luminosita INT(4))
```

#	Nome	Tipo	Collation	Attributi	Null	Predefinito	Extra
<input type="checkbox"/>	1 <b>id</b>	int(10)			No	<i>Nessuno</i>	AUTO_INCREMENT
<input type="checkbox"/>	2 <b>stazione</b>	int(3)			Sì	<i>NULL</i>	
<input type="checkbox"/>	3 <b>dataora</b>	datetime			Sì	<i>NULL</i>	
<input type="checkbox"/>	4 <b>luminosita</b>	int(4)			Sì	<i>NULL</i>	

### Codice "salvaluminosita.php"

```
<?php
$stazione=$_GET['stazione'];
$luminosita=$_GET['luminosita'];
$ora = date("Y-m-d H:i:s");
$conn=mysqli_connect("myhost","myusername","mypassword","stazioni");
/* $conn=mysqli_connect("192.18.1.3","root","mypassword","stazioni");
   se utilizzo un Server locale*/
$s="Insert into misure (stazione,dataora,luminosita) values
($stazione,'$ora',$luminosita)";
$q=mysqli_query($conn,$s);
mysqli_close($conn);
```

?>

id	stazione	dataora	luminosita
484	1	2021-04-23 15:14:02	907
485	1	2021-04-23 15:24:02	709
486	1	2021-04-23 15:34:03	788
487	1	2021-04-23 15:44:03	724
488	1	2021-04-23 15:54:03	766
489	1	2021-04-23 16:04:04	744
490	1	2021-04-23 16:14:04	755
491	1	2021-04-23 16:24:04	825
492	1	2021-04-23 16:34:04	812
493	1	2021-04-23 16:44:05	819
494	1	2021-04-23 16:54:05	763
495	1	2021-04-23 17:04:05	744
496	1	2021-04-23 17:14:06	726
497	1	2021-04-23 17:24:06	779
498	1	2021-04-23 17:34:06	791
499	1	2021-04-23 17:44:06	756
500	1	2021-04-23 17:54:07	855
501	1	2021-04-23 18:04:07	834
502	1	2021-04-23 18:14:07	754

## Visualizzazione dei dati

La pagina [www.maurodeberardis.it/CodiciPHP/displaydati.php](http://www.maurodeberardis.it/CodiciPHP/displaydati.php) (o [localhost/retestazioni/displaydati.php](http://localhost/retestazioni/displaydati.php)) visualizza i dati memorizzati nella tabella "misure" del database "stazioni". La pagina, molto semplice ed essenziale, stampa in una tabella le misure, filtrate eventualmente per stazione di monitoraggio, di una determinata giornata.

### Codice "displaydati.php"

```
<!DOCTYPE html>
<html>
<head><title>Rete di stazioni di monitoraggio Arduino</title></head>
<style>
  td,th{border:1px solid #aaaaaa;text-align:center}
</style>
<script language="JavaScript">
function valida()
{
    dr=document.form1.data_ril.value;
    if(dr=="")
    {
        alert("Inserire la data");
        return false;
    }
}
</script>
<body>
<?php
if(!isset($_GET["submit"]))
{
    ?>
    <form name="form1" onsubmit="return valida()" >
        <h2>Rete di stazioni di monitoraggio Arduino</h2>
        <label>Data di rilevamento</label><br/>
        <input type="date" id="data_ril" name="data_ril">
        <br/><br/><label>Stazioni di monitoraggio</label>
        <ul>
        <li><input type="radio" name="stazioni" value="T" checked>
        Tutte le stazioni</li>
        <li><input type="radio" name="stazioni" value="1">
        Stazione 1 - Viale Europa Teramo</li>
        <li><input type="radio" name="stazioni" value="2">
        Stazione 2 - Piazza Dante Teramo</li>
        </ul><br/>
        <input type="submit" name="submit" value="Display dati">
        <input type="reset" value="Reset">
    </form>
<?php
}
else
{
    $data_ril= $_GET['data_ril'];
    $stazioni=$_GET['stazioni'];
    $aa=substr($data_ril,0,4);
    $mm=substr($data_ril,5,2);
    $gg=substr($data_ril,8,2);
    $data=$gg."/".$mm."/".$aa;
    $aa=(int)$aa; $mm=(int)$mm; $gg=(int)$gg;
```

```

$titleo="Data rilevamento: ".$data." - " ;
switch($stazioni)
{
    case "T":
        $filtrostazioni=" stazione>0 ";
        $titleo.="Tutte le stazioni";
        break;
    case "1":
        $filtrostazioni=" stazione=1 ";
        $titleo.="Stazione 1";
        break;
    case "2":
        $filtrostazioni=" stazione=2 ";
        $titleo.="Stazione 2";
        break;
    default:
}
echo"<h2>Rete di stazioni di monitoraggio Arduino</h2>";
echo"<h2>$titleo</h2>";
$conn=mysqli_connect("myhost","myusername","mypassword","stazioni");
/* $conn=mysqli_connect("localhost","root","mypassword","stazioni");
   se utilizzo un Server locale*/
$s="Select * from misure where
Year(dataora)=$aa And Month(dataora)=$mm And Day(dataora)=$gg
And $filtrostazioni
order by dataora Asc";
$q=mysqli_query($conn,$s);
(nr=mysqli_num_rows($q);
if ($nr==0)
{
    echo "<br><br>Nessun dato trovato<br><br>";
}
else
{
    echo "<table width=30%>";
    echo "<tr><th>Nr</th><th>Stazione</th>
        <th>Time misura</th><th>Luminosita'</th>
        </tr>";
    $k=0;
    while($r=mysqli_fetch_array($q)
    {
        $k++;
        $timemisura=substr($r[2],11,8);
        echo "<tr><td>$k</td>
            <td>$r[1]</td><td>$timemisura</td>
            <td>$r[3]</td></tr>";
    }
    echo"</table>";
}
mysqli_close($conn);
echo"<br><br><a href='displaydati.php'>Scegli un'altra data</a>";
}
?>
</body>
</html>

```

Test con server remoto: [www.maurodeberardis.it/CodiciPHp/displaydati.php](http://www.maurodeberardis.it/CodiciPHp/displaydati.php)

## Rete di stazioni di monitoraggio Arduino

Data di rilevamento

23/04/2021 

Stazioni di monitoraggio

- Tutte le stazioni
- Stazione 1 - Viale Europa Teramo
- Stazione 2 - Piazza Dante Teramo

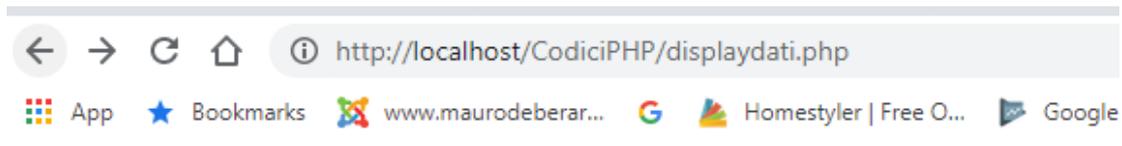
## Rete di stazioni di monitoraggio Arduino

Data rilevamento: 23/04/2021 - Stazione 1

Nr	Stazione	Time misura	Luminosita'
1	1	15:14:02	907
2	1	15:24:02	709
3	1	15:34:03	788
4	1	15:44:03	724
5	1	15:54:03	766
6	1	16:04:04	744
7	1	16:14:04	755
8	1	16:24:04	825
9	1	16:34:04	812
10	1	16:44:05	819
11	1	16:54:05	763
12	1	17:04:05	744
13	1	17:14:06	726
14	1	17:24:06	779
15	1	17:34:06	791
16	1	17:44:06	756
17	1	17:54:07	855
18	1	18:04:07	834

[Scegli un'altra data](#)

## Test con Server locale: localhost/CodiciPHP/displaydati.php



# Rete di stazioni di monitoraggio Arduino

Data di rilevamento

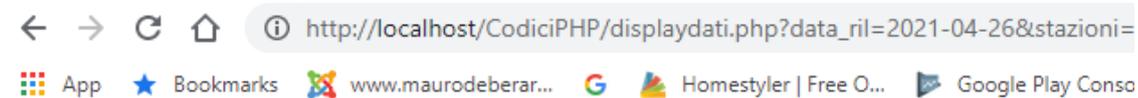
26/04/2021 

Stazioni di monitoraggio

- Tutte le stazioni
- Stazione 1 - Viale Europa Teramo
- Stazione 2 - Piazza Dante Teramo

Display dati

Reset



# Rete di stazioni di monitoraggio Arduino

## Data rilevamento: 26/04/2021 - Tutte le stazioni

Nr	Stazione	Time misura	Luminosita'
1	1	19:03:53	730
2	2	19:04:08	910
3	1	19:13:54	571
4	2	19:14:08	905
5	1	19:23:54	422
6	2	19:24:09	652
7	1	19:33:54	450
8	2	19:34:09	642

[Scegli un'altra data](#)