

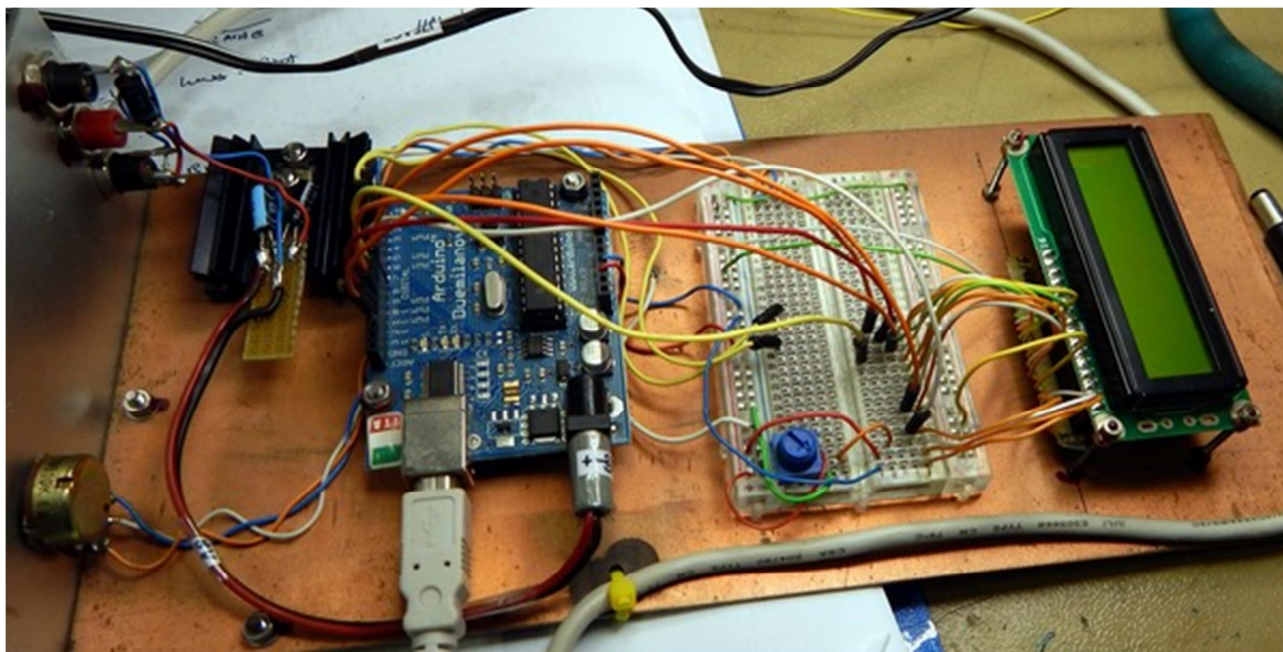
UTILIZZO DISPLAY 16X2 SU BREAD-BOARD ARDUINO

RIFERIMENTI

<i>Genere</i>	<i>DATA</i>	<i>Generalità</i>	<i>Note</i>	<i>Distribuzione</i>
<i>radio</i>	<i>Feb 16</i>	<i>laboratorio</i>		<i>af</i>

GENERALITA'

Utilizzare ARDUINO con efficacia per prove in laboratorio richiede un minimo di infrastruttura. Ho destinato una ormai vecchia piastrina del modello DUEMILANOVE che ho piazzato solidamente su di una piastra di vetronite ramata, insieme con un regolatore ad 8 Volt per alimentazione da esterno senza il PC, una piccola bread-board per implementare circuiti aggiuntivi ed un display da 16 caratteri x due righe.



In aggiunta ci sono un potenziometro da 10kOhm lineare con manopola e la protezione sui morsetti di alimentazione esterna, che può andare da 10 a 15 Volt, sia con banana jack che con le prese comuni per alimentazione.

Senza questo minimo set di stabilità mi ritrovavo più volte a cercare i fantasmi per via che un filo della parrucca si era staccato accidentalmente, anche con circuiti molto semplici.

Una ulteriore miglioria è stata il collegare il display interamente agli ultimi pin digitali di ARDUINO, da 8 a 13, in modo da avere liberi gli altri pin per giocare.

SCHEMA ELETTRICO

Un regolatore con 7808 TO220 su dissipatore può alimentare ARDUINO dall'esterno.

La piastra prototipi e il display sono alimentati con il 5V di ARDUINO stesso. Quindi la circuiteria esterna è limitata alle possibilità di ARDUINO stesso di alimentare. Essendo però un sistema chiuso sono limitati i rischi di rottura di ingressi per tensioni errate.

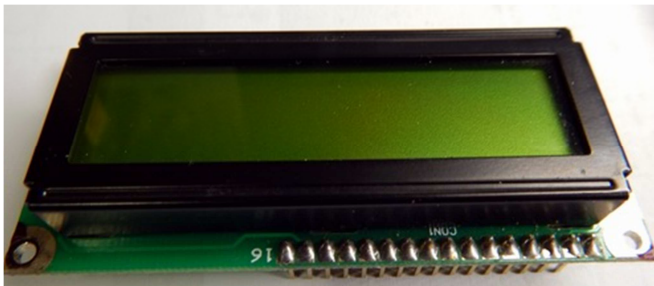
Il display utilizzato permette il comando via 4 linee dati più i controlli. Il controllo ENABLE viene comandato dal digitale 13 che monta un led sulla scheda, quindi sta normalmente spento e lampeggia quando il display viene comandato.

Il controllo R/W sta a zero fisso permettendo di inviare dati al display, ma non di riceverne.



Il circuito di regolazione del contrasto ha in totale circa 10kOhm, ed il potenziometro a trimmer è da 2k. Il range di regolazione sul pin C del display va da 0 (contrasto massimo) a circa +1.5V (contrasto minimo).

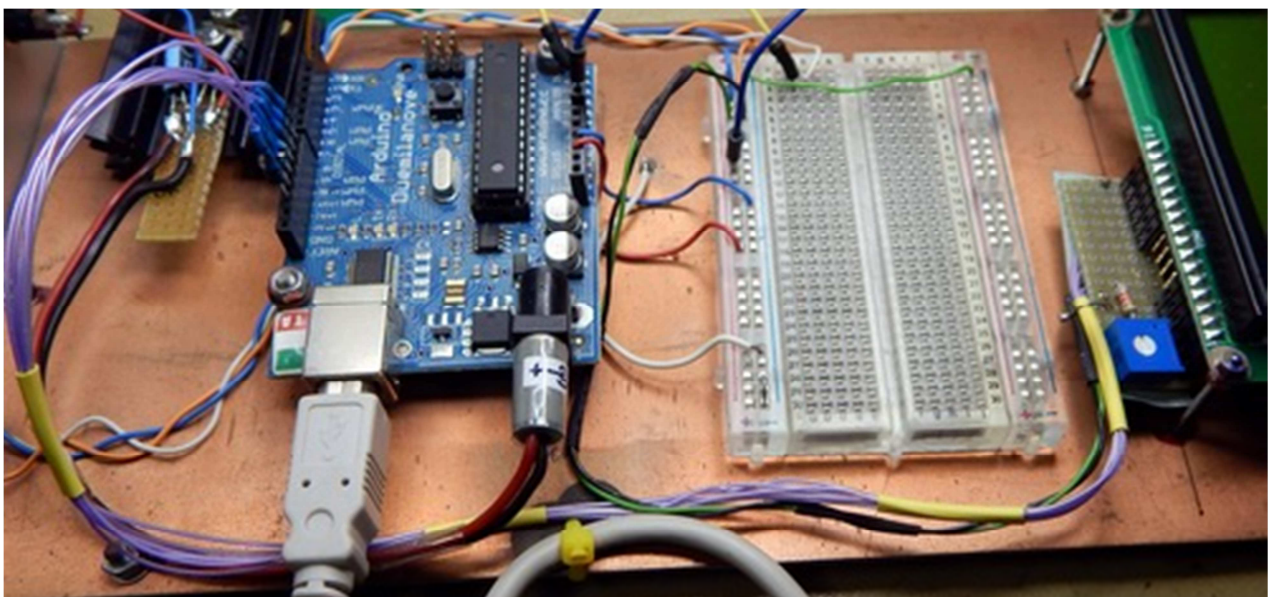
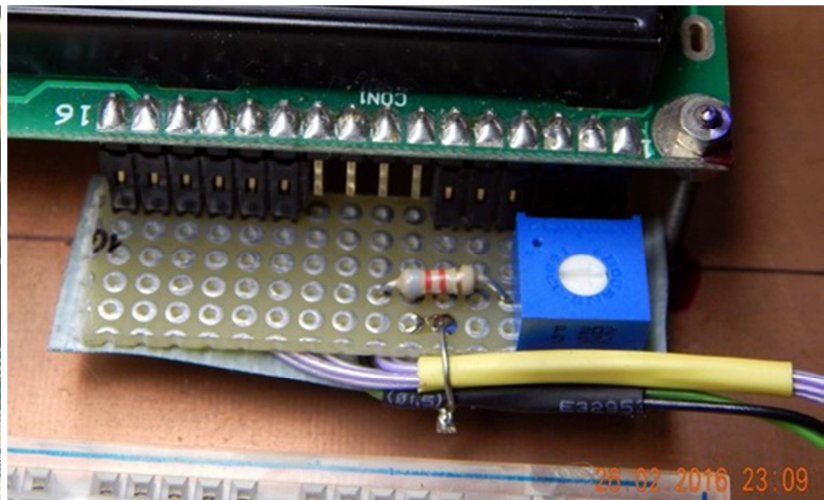
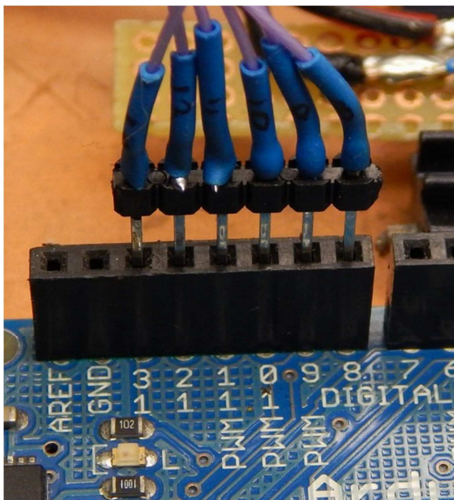
La retro illuminazione del display è presa dal 5V stesso, non è molto forte e porta un consumo, spostarla su alimentazione indipendente potrebbe essere una miglioria futura.



ESECUZIONE

Unica nota il cavetto di collegamento al display: utilizza fili isolati in teflon con un buon rapporto sezione rame/isolante, quindi è molto fine.

Il resto è convenzionale.





SKETCH

Riporto il listato di uno sketch che utilizza il display con i collegamenti illustrati, per riferimento. Le istruzioni che hanno un senso con questo circuito sono evidenziate. Le altre si riferiscono ad un altro progetto e sono una versione di prova da non copiare.

```
// include the library code:
#include <LiquidCrystal.h>

unsigned long time;
unsigned long previousMillis = 0;          // will store last time LED was updated
const long interval = 1000;  // interval at which to blink (milliseconds)
unsigned long currentMillis = 0;

// initialize the library with the numbers of the interface pins
LiquidCrystal lcd(8, 13, 9, 10, 11, 12);

byte secnd = 0; //per caricamento array
byte i = 0; //indice generico
int sensorvac = A0;    // select the input pin for the An in vac
int rawVac = 0;  // variable to store the raw value vac
int Vac;  //Vac rms
int sensorcur = A1;    // select the input pin for the An in vac
int rawcur = 0;  // variable to store the raw value vac
int cur;  //cur rms
float watt = 0;
float amp = 0;
float pwr1[60];
float pwr2[60];
int vlt1[60];
int vlt2[60];
float amp1[60];
float amp2[60];

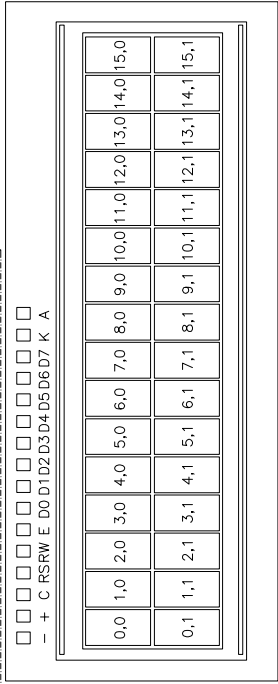
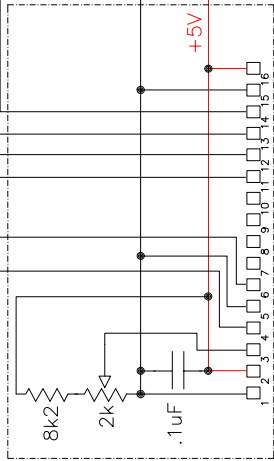
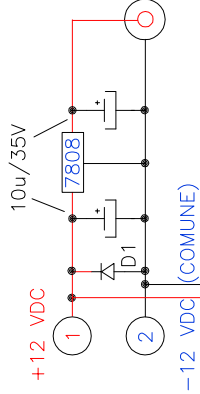
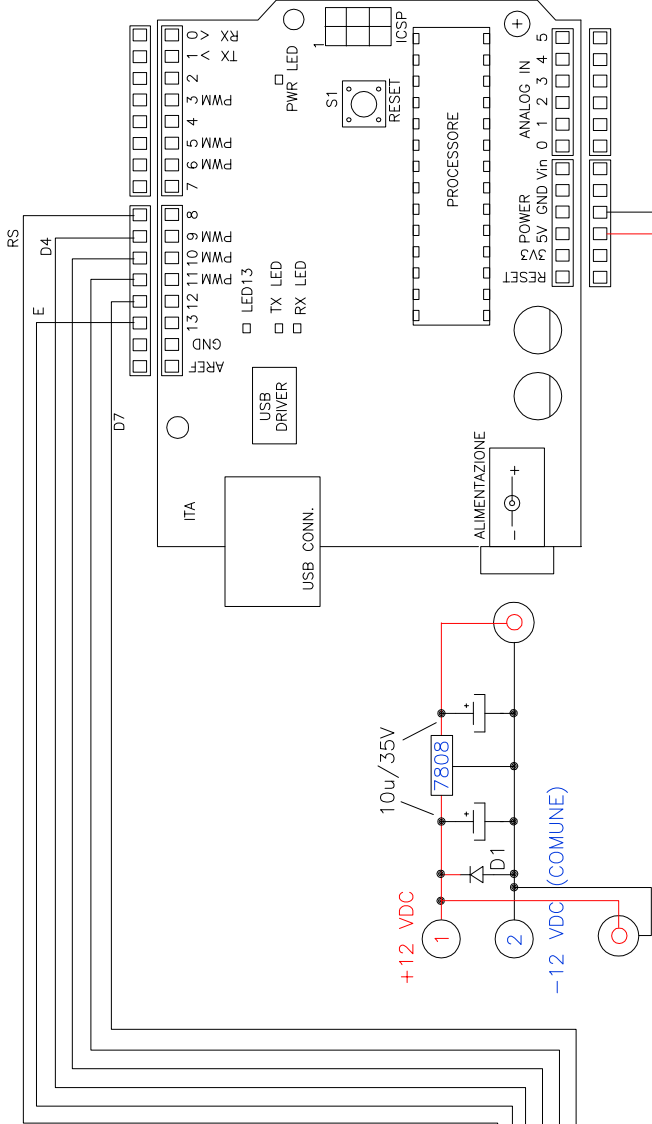
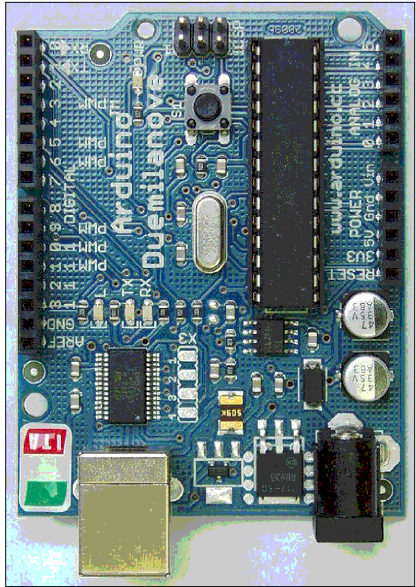
void setup() {
  Serial.begin(9600);
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  analogReference(DEFAULT); //reference adc a 5V internal
}

void loop() {
  //timing area
  time = millis();
  currentMillis = time;
  if (currentMillis - previousMillis >= interval) { //primo if
    previousMillis = currentMillis;
  }
  //acq area
  // read the value from the sensor:
  rawVac = analogRead(sensorvac);
  Vac=map(rawVac,0,1023,170,250);
  rawcur = analogRead(sensorcur);
```



```
cur = map(rawcur,0,1023,0,20000);
amp = (float)cur / 1000.0;
watt = (float)Vac * amp;
watt = (float)watt / 1000.0;
//array area
pwr1[secnd] = watt;
vlt1[secnd] = Vac;
amp1[secnd] = amp;
//secnd++;
if (secnd == 59) {
    secnd = 0;
} else {
    secnd++;
}
//serial com area
Serial.print("Time: ");
//prints time since program started
Serial.println(time);
for (i=0;i<60;i=i+1){
    Serial.print(pwr1[i]);
    Serial.println(";");
}
//display area
// lcd.setCursor set the cursor to column (0-15), line (0-1)
//prima riga kw
lcd.setCursor(0, 0);
lcd.print("kVA ist.=");
lcd.setCursor(9, 0);
lcd.print(watt);
//da acq
lcd.setCursor(0, 1);
lcd.print("Vac=");
lcd.setCursor(4, 1);
lcd.print(Vac); //è allineato a sx
lcd.setCursor(8, 1);
lcd.print("mA=");
lcd.setCursor(11, 1);
lcd.print(cur);
lcd.print(" ");
} //fine primo if
} //fine loop
```

Buon divertimento, Alessandro Frezzotti



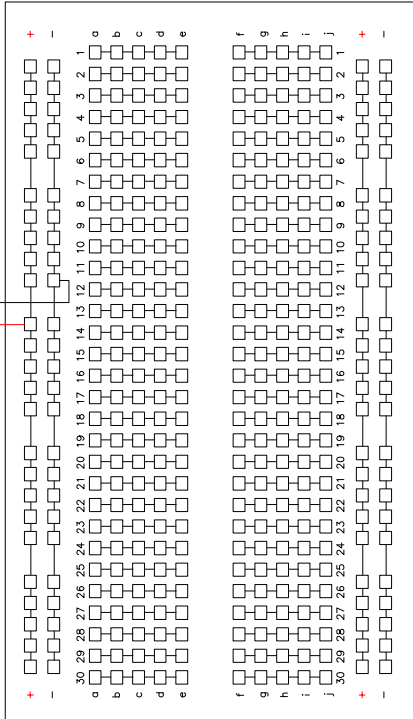
```
comandi/istruzioni
#include <LiquidCrystal.h>
LiquidCrystal lcd(8,13,9,10,11,12);
lcd.begin(16,2); // in setup

DISPLAY 16 CH X 2 ROW
1210 353660 02
C216L01YBW00
```

C IN MICROF DOVE NON INDICATO
R IN OHM DOVE NON INDICATO

Questo documento e' da intendersi RISERVATO. La sua riproduzione anche parziale o la sua
cessione a terzi deve essere espressamente autorizzata da ALESSANDRO FREZZOTTI - IZ5AGZ

displ16x2_0.DWG



piastro prototipo

10k linear