

```
#include <LiquidCrystal.h> //inclusione della libreria LCD
```

```
// definizione dei pin
```

```
int ACS714 = A0;
```

```
int partitore = A1;
```

```
//definizione valori integer e float per calcolo corrente
```

```
int sensorValue = 0;
```

```
int outputValue = 0;
```

```
float amps = 0.0;
```

```
float vref = 5.0;
```

```
//definizione valori integer e float per calcolo tensione
```

```
int valV = 0;
```

```
float pinVoltage = 0;
```

```
float volts = 0.0;
```

```
float ratio = 2.84;
```

```
//definizione dei pin LCD
```

```
int RS = 5;
```

```
int EN = 4;
```

```
int D7 = 0;
```

```
int D6 = 1;
```

```
int D5 = 2;
```

```
int D4 = 3;
```

```
//inizializzazione libreria LCD
```

```
LiquidCrystal lcd(RS,EN,D4,D5,D6,D7);
```

```
void setup(){
```

```
    analogRead(ACS714); // prima lettura analogica della corrente
```

```
    analogRead(partitore); // prima lettura analogica della tensione
```

```
    lcd.begin(16,2); //inizializzazione lcd 16x2
```

```
    lcd.clear(); // pulisci memoria lcd
```

```
}
```

```
void loop(){
```

```
    sensorValue=analogRead(ACS714); //da adesso la lettura del sensore ACS714 verrà richiamata dalla stringa sensorValue
```

```
    outputValue = (((long)sensorValue*5000/1024)-500)*1000/66; //calcolo della corrente in base alla tensione in uscita dall'ACS714 e restituzione valore in milliampere
```

```
    amps=(float)outputValue/1000; //divisione del valore in mA per 1000 per restituzione del valore in Ampere e richiamo sotto la stringa amps
```

```
    valV = analogRead(partitore); //da adesso la lettura del partitore verrà richiamata dalla stringa valV
```

```
    pinVoltage = valV*0.00488; // ogni step rilevato in ingresso vale 0.00488V facendo 5.0V/1024, si ha quindi il calcolo degli step dell'ADC del uController per la relativa tensione reale
```

```
    volts = pinVoltage * ratio; // calcola la tensione reale moltiplicando quella in ingresso all'ADC per il ratio del partitore resistivo, di seguito la spiegazione per il calcolo del valore ratio
```

```
    /*
```

Per il valore ratio va calcolato prima il partitore in base alle tensioni volute per fare in modo che abbia in uscita 5V accettati da arduino. dopo di che collegare al partitore una

tensione di riferimento pari alla massima applicabile in ingresso misurare quindi esattamente la tensione in ingresso al partitore e quindi in uscita tra il nodo delle 2 resistenze del partitore

e GND, fare il calcolo dividendo la prima tensione misurata diviso la seconda, ad esempio se diamo in ingresso una tensione di 30V/5V avremo un ratio di 6.0 ma non è esatto in quanto abbiamo

utilizzato dei valori inesatti per via della tolleranza delle resistenze....pertanto se avremo 30V in ingresso e 4.8 in uscita avremo un ratio di 6.25. per sicurezza è consigliabile

mettere uno zener da 5.1 tra l'ingresso analogico di arduino e il partitore in modo da evitare sovratensioni che potrebbero danneggiare l'ingresso.

Per una tensione in ingresso di 30V è consigliabile utilizzare lo schema allegato in paint.

*/

```
lcd.setCursor(3,0); //setta il cursore sul 4° carattere
```

```
lcd.print("AMPS: "); //scrivi la parola AMPS:
```

```
lcd.print(amps); // riporta il valore calcolato degli ampere
```

```
lcd.setCursor(3,0); // setta il cursore sul 4° carattere
```

```
lcd.print("VOLT: "); // scrivi la parola VOLTS:
```

```
lcd.print(volts); // riporta il valore calcolato dei volts
```

```
}
```

