

Il software

Nelle righe di codice è riportato un estratto delle parti rilevanti del software realizzato per la gestione dell'analizzatore.

```
/*  
  
    Battery Monitor  
  
    by Antonio Viggiani (2013) versione alfa  
  
*/  
  
#define AVG_NUM 32          // numero di iterazioni per la lettura media  
#define BS_VOLT_CHAN 2     // adc pin di lettura analogica  
  
float CAPACITA = 200.0;    // capacita nominale batteria (C20)  
float amper_ora = CAPACITA; // capacita residua batteria  
float k = 1.4;             // esponente di Peukert  
float rendimentoRicarica = 0.95; // efficienza in ricarica  
  
/* funzione per la lettura della tensione */  
int read_adc(int channel){  
  
    int sum = 0;  
  
    int temp = 0;  
  
    int i;  
  
    for (i=0; i<AVG_NUM; i++) {  
  
        temp = analogRead(channel);  
  
        sum += temp;  
  
    }  
  
    return(sum / AVG_NUM);  
  
}  
  
/* funzione per la lettura della corrente */  
float read_current(){  
  
    float sum = 0.0;  
  
    float temp;
```

```

int i;

for (i=0; i<AVG_NUM; i++) {    // loop through reading raw adc values AVG_NUM number of times

    temp = monitor.shuntCurrent() * BS_AMP_PER_SCALE + amper_offset;    // read the input INA219

    sum += temp;                // store sum for averaging

}

return(sum / AVG_NUM);        // divide sum by AVG_NUM to get average and return it

}

```

/* funzione per la lettura della tensione di riferimento dell'Arduino */

```

long readVcc() {

    long result;

    // Read 1.1V reference against AVcc

    ADMUX = _BV(REFS0) | _BV(MUX3) | _BV(MUX2) | _BV(MUX1);

    attendi(2); // Wait for Vref to settle

    ADCSRA |= _BV(ADSC); // Convert

    while (bit_is_set(ADCSRA,ADSC));

    result = ADCL;

    result |= ADCH<<8;

    result = 1126400L / result; // Back-calculate AVcc in mV

    return result;

}

```

/* funzione per il calcolo della corrente di Peukert*/

```

float CurrentPeukert(float c){

    float temp = 0.0;

    float temp1=0.0;

    temp = -1.0*c;

    temp1 = (pow(temp, k)/pow(CAPACITA/20.0, k-1.0)) * -1.0;

    if (temp1 * -1.0 < temp)

    {

        temp1 = temp * -1.0;

    }

}

```

```

return temp1;
}

void setup() {
// inilizzazione
}

void loop() {
float temp;

temp = read_adc(BS_VOLT_CHAN);

voltBS = temp/1023.0*readVcc()/1000*5.0;

Current = read_current();

time = millis();

intervallo_tempo = (time - old_time)/1000.0/3600.0;

old_time = time;

if (Current >= 0.0 && (amper_ora < CAPACITA))
{
amper_ora = (intervallo_tempo * Current) * rendimentoRicarica + old_amper_ora;
}
else
{
if (Current < 0.0)
{
amper_ora = (intervallo_tempo * CurrentPeukert(Current)) + old_amper_ora;
}
}
old_amper_ora = amper_ora;
}

```

Appendice

La legge di Peukert

La legge di Peukert è una semplice relazione che mette lega la capacità di una batteria al Piombo in funzione dell'intensità della corrente prelevata. Come quest'ultima cresce e più decresce la capacità della batteria.

Normalmente la capacità riportata dai costruttori come dato di targa è quella ottenibile con una corrente di scarica pari ad un ventesimo (C20) della sua capacità, ad esempio per una batteria da 100 Ah significa che la stessa è in grado di fornire una corrente di 5 A per 20 ore.

Per correnti superiori la capacità della batteria si riduce secondo la legge di Peukert:

$$C_p = I^k \cdot t$$

dove

C_p è la capacità espressa in Ah,

I è l'intensità della corrente di scarica,

t è il tempo in ore della scarica.

Si ottiene ad esempio che per una batteria 100 Ah sottoposta a una corrente di scarica di 10 A la sua durata non sarà pari a 10 h ma circa a 8,7 A, in pratica è come se la capacità della batteria si fosse ridotta a circa 87 A.

La legge di Peukert può anche essere scritta come:

$$It = C \cdot \left(\frac{C}{I \cdot H} \right)^{k-1}$$