

RELAZIONE DI SPETTROSCOPIA

Camera di Bragg

FRANCESCO FORCHER

Università di Padova, Facoltà di Fisica
francesco.forcher@studenti.unipd.it
Matricola: 1073458

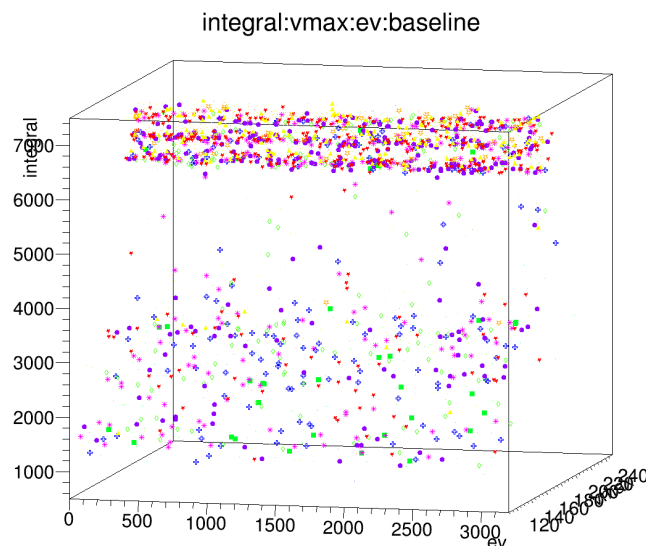
ENRICO LUSIANI

Università di Padova, Facoltà di Fisica
enrico.lusiani@studenti.unipd.it
Matricola: 1073300

LAURA BUONINCONTRI

Università di Padova, Facoltà di Fisica
laura.buonincontri@studenti.unipd.it
Matricola: 1073131

4 luglio 2016



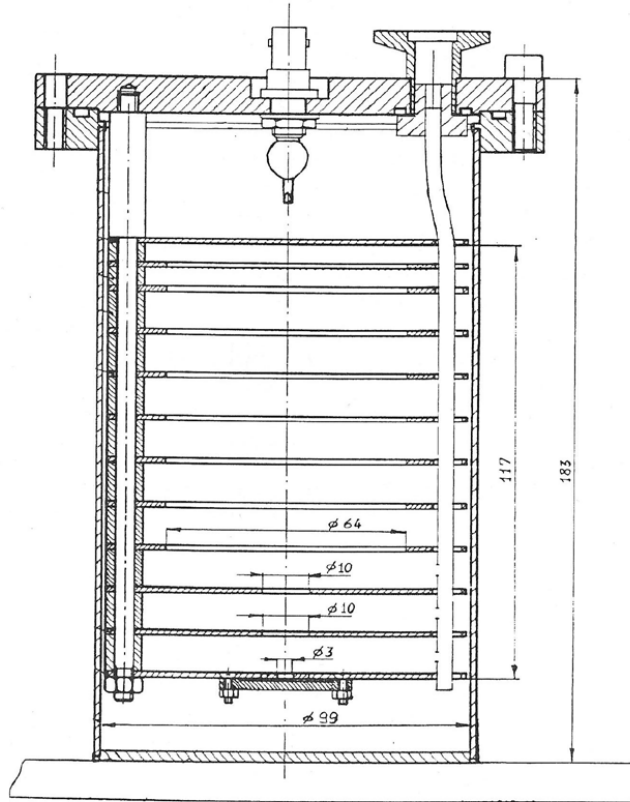
Sommario

- *Studio della risposta alla radiazione alfa di una camera di Bragg*
- *Analisi dei segnali per calcolare l'energia depositata e l'altezza del picco di Bragg*
- *Studio della risposta al variare dell'energia della particella e al variare della pressione*
- *Misurazione del range delle particelle e verifica della relazione tra range e pressione*

INDICE

I Schema camera	4
II Parte I	4
I Prima misura delle particelle alfa	4
II Risoluzione energetica	10
III Parte II	11
I Misure a pressione 650mb	12
II Misure a pressione 550mb	15
III Misure a pressione 500mb	19
IV Misure a pressione 450mb	23
V Misure a pressione 400mb	27
VI Misure a pressione 380mb	31
VII Curve in funzione della pressione	34
IV Tabelle	43
V Discussioni e conclusioni	43
VI Codice	44

I. SCHEMA CAMERA



II. PARTE I

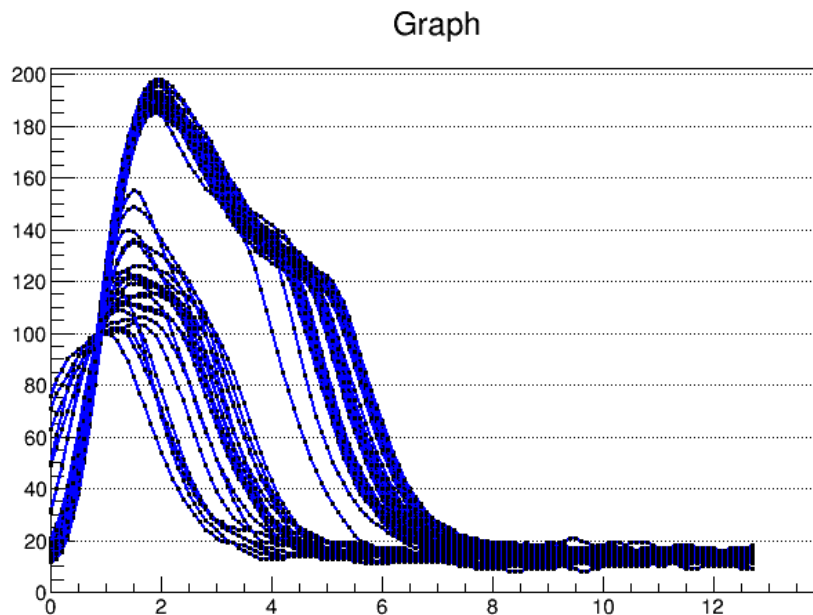
II.1 Prima misura delle particelle alfa

Sono stati settati gli strumenti, a pressione 600 mb. Si è mantenuto lo Shaping Time a 0.25-0.5 μ s, dopo aver osservato il comportamento. E' stata regolata l'amplificazione in modo da mantenere il picco attorno ai 3V. E' stato impostato il trigger in modo tale che fosse circa a metà altezza del picco sull'oscilloscopio.

Si è verificato che il numero di segnali spuri fosse inferiore al 30% del totale, facendo il grafico dei picchi e calcolando l'integrale dei segnali a bassa energia.

E' stato quindi acquisito il primo set di dati (circa 3000 eventi)

E' stato acquisito anche un set di dati con meno eventi e un trigger molto più basso per stimare la baseline.

Grafico 1 Grafico segnali a 600mb in funzione del tempo (μs)

E' stato fatto il grafico degli integrali a partire dal file a basso trigger, dove si è visto un picco a bassa energia, corrispondente agli eventi di fondo, e selezionando questi eventi è stato disegnato un istogramma della variabile baseline, da cui calcolando il centroide del picco principale, è stato ricavato il valore da inserire nella macro al posto del valore di default.

L'analisi dei dati che segue è stata effettuata utilizzando la macro fornite dal laboratorio, modificando il limite dei campioni da integrare e inserendo il valore della baseline appena stimato. Il limite dei campioni a 600 mb è stato posto uguale a 90.

Grafico 2 Grafico segnali baseline in funzione del tempo (μs)

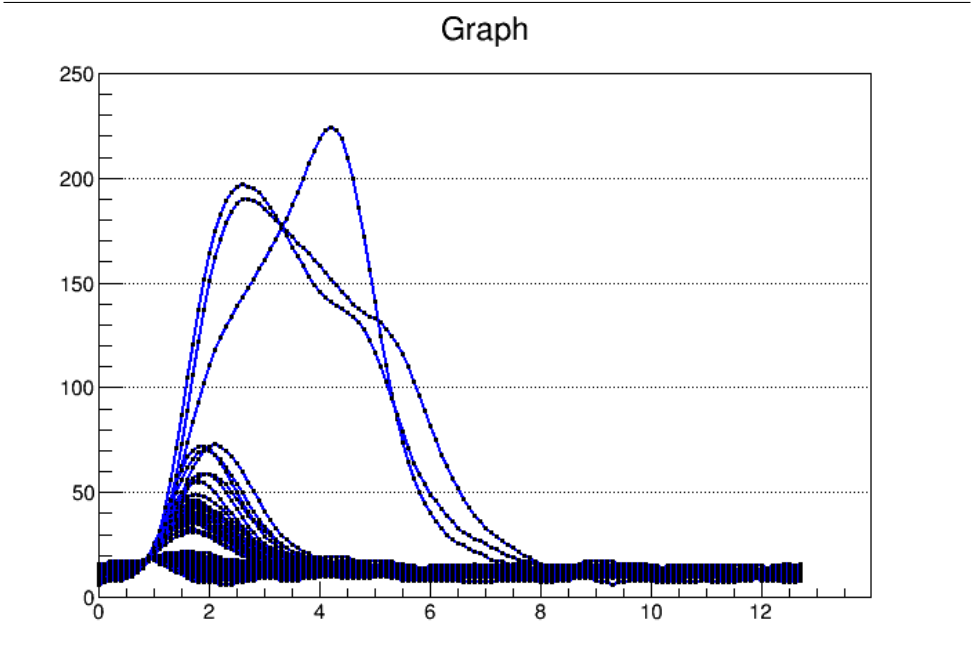


Grafico 3 Picco della baseline

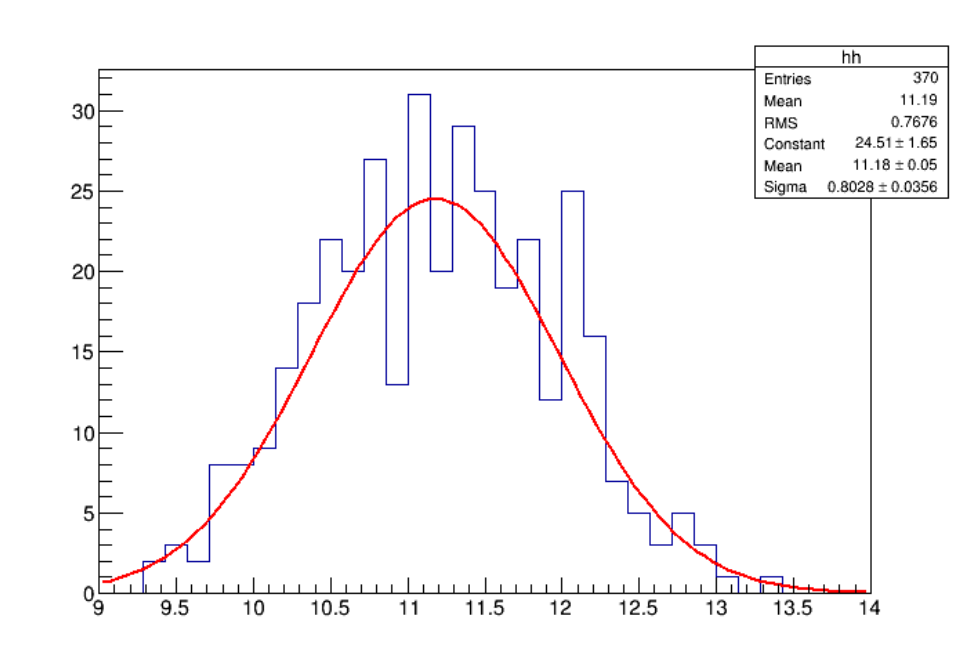


Grafico 4 Grafico integral

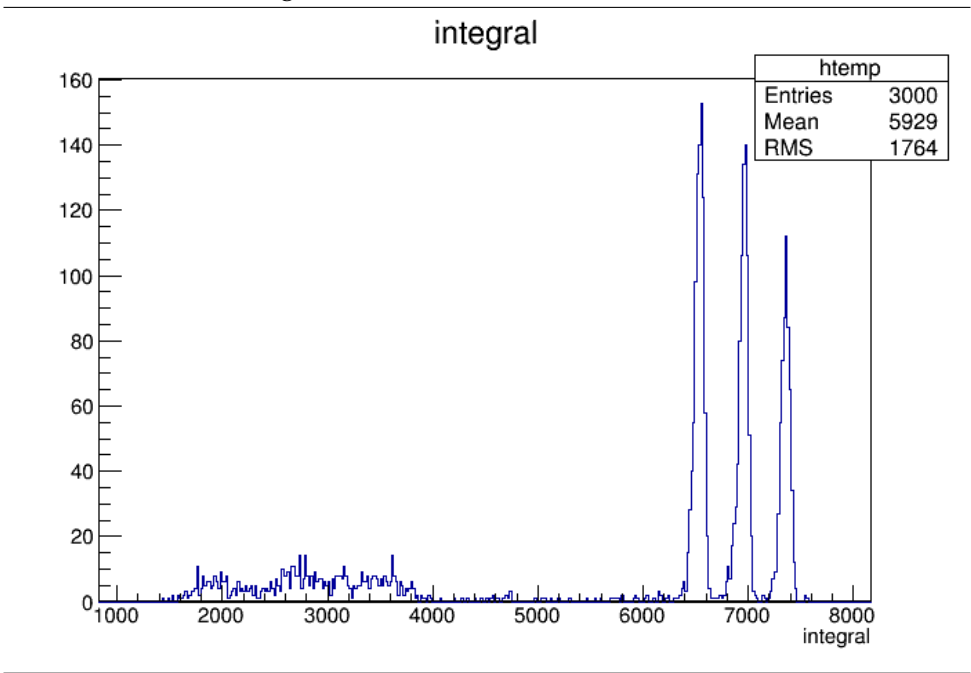


Grafico 5 Grafico integral:ev

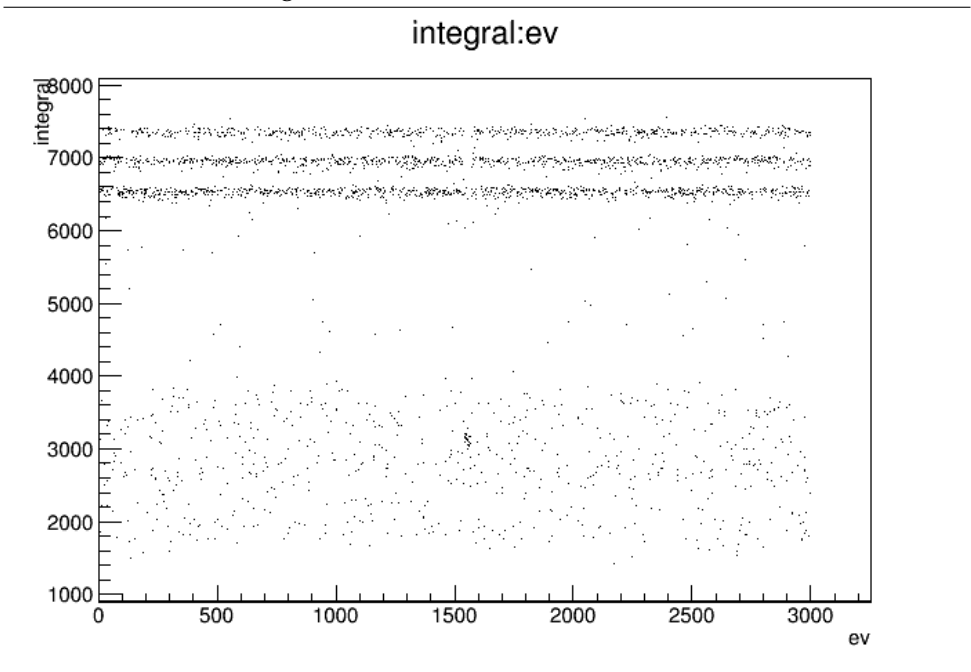


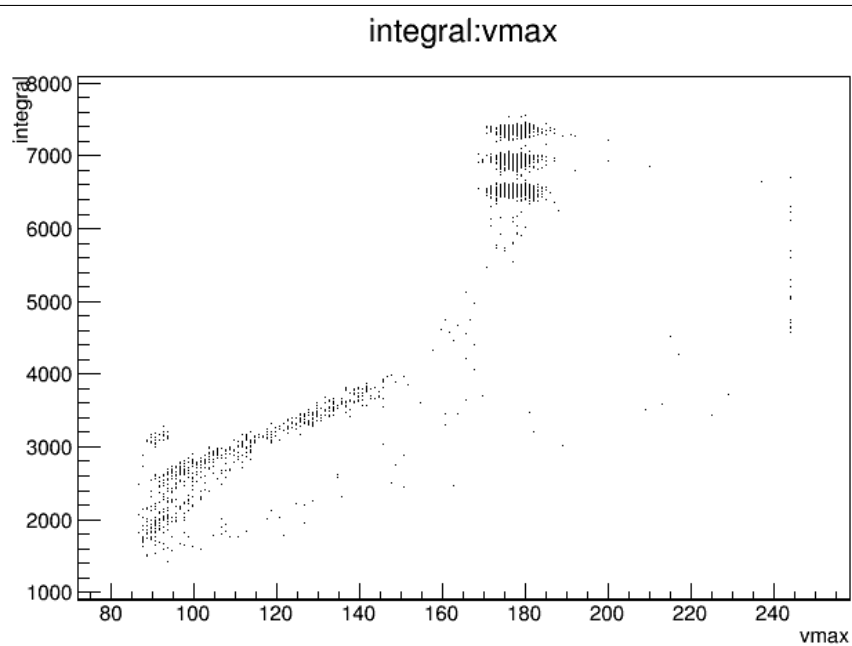
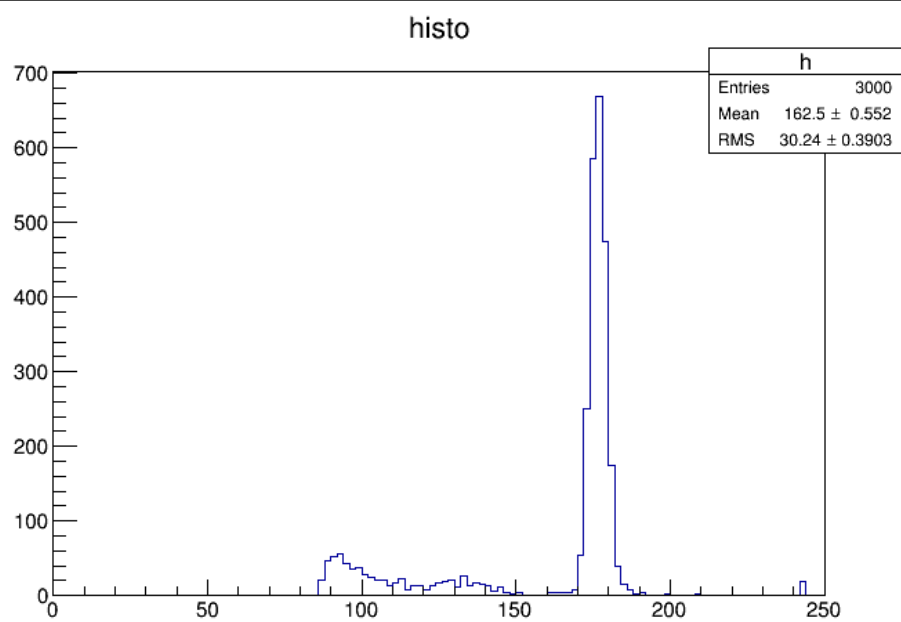
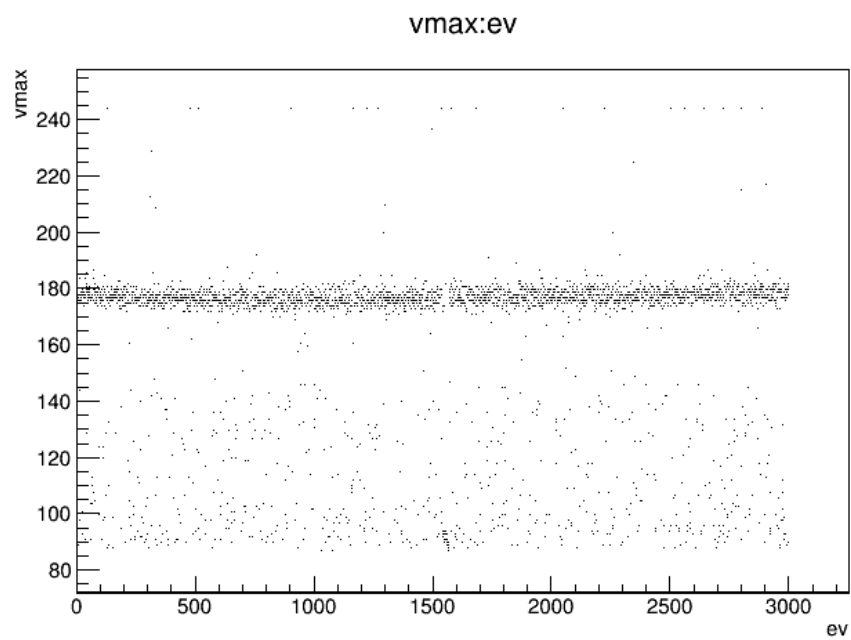
Grafico 6 Grafico integral:vmax**Grafico 7** Grafico vmax

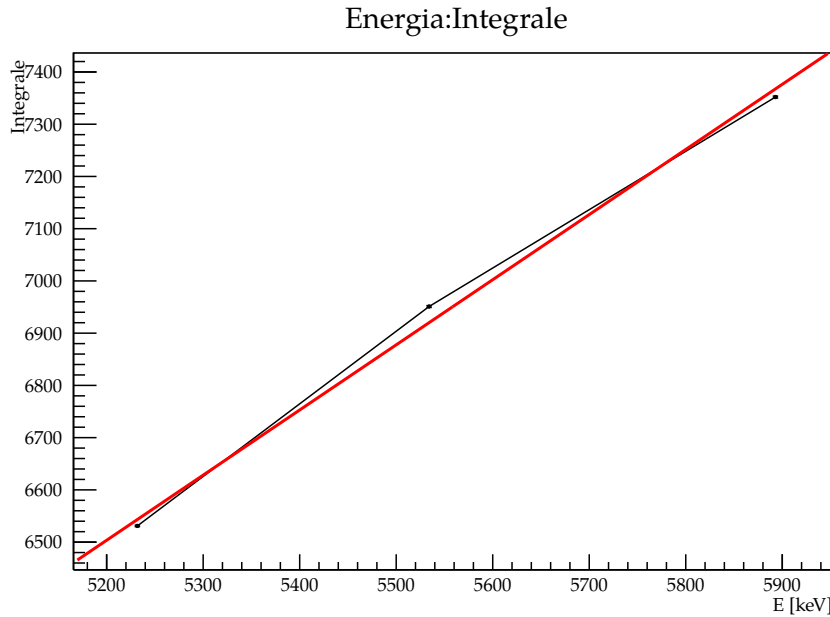
Grafico 8 Grafico vmax:ev

II.II Risoluzione energetica

Per il calcolo della risoluzione energetica, abbiamo per prima cosa trovata la relazione tra integrale ed energia, ipotizzandola lineare. Per fare ciò abbiamo calcolato l'energia teorica di ciascun picco, facendo la media delle energie dei decadimenti alfa, pesate sulla loro probabilità. Poi abbiamo calcolato l'integrale relativo a ciascun picco, come centroide del picco nell'istogramma degli integrali, e il suo errore, l'RMS del picco. Da questi dati abbiamo proceduto ad un'interpolazione dell'integrale in funzione dell'energia, ricavando così la funzione energia:integrale. Usando la funzione, abbiamo riscritto l'asse delle ascisse nell'istogramma degli integrali, in modo che mostrasse l'energia. Da questo nuovo istogramma abbiamo ricavato la risoluzione energetica, misurando l'RMS dei picchi e usandola per la formula

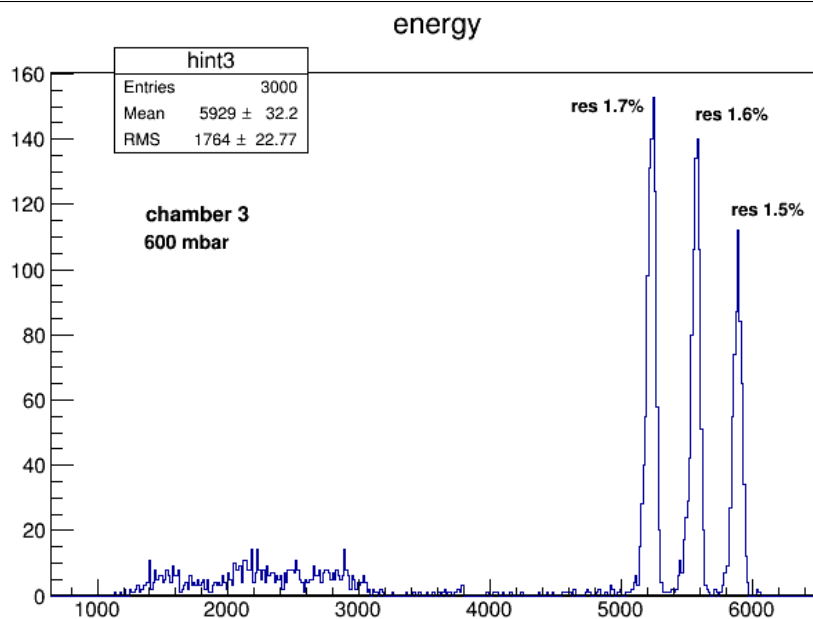
$$R = \frac{FWHM_E}{E} = \frac{2.335 \cdot \sigma_E}{E} \quad (1)$$

Grafico 9 Grafico Energia:Integrale



$$q = 19.6792 \pm 19.5559$$

$$m = 1.2469 \pm 0.00354684$$

Grafico 10 Risoluzioni energetiche, grafico Energia(keV):conteggio

III. PARTE II

L'analisi dei dati che segue è stata effettuata utilizzando la macro fornita dal laboratorio, modificando il limite dei campioni e inserendo il valore della baseline stimato in precedenza.

III.I Misure a pressione 650mb

Grafico 11 Grafico segnali a 650mb

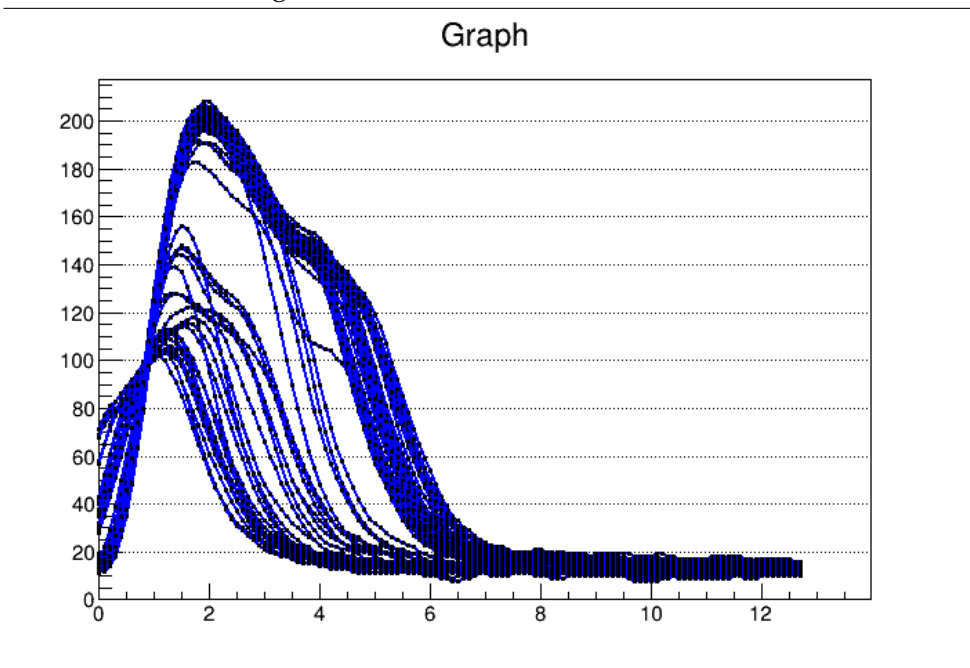


Grafico 12 Grafico integrale

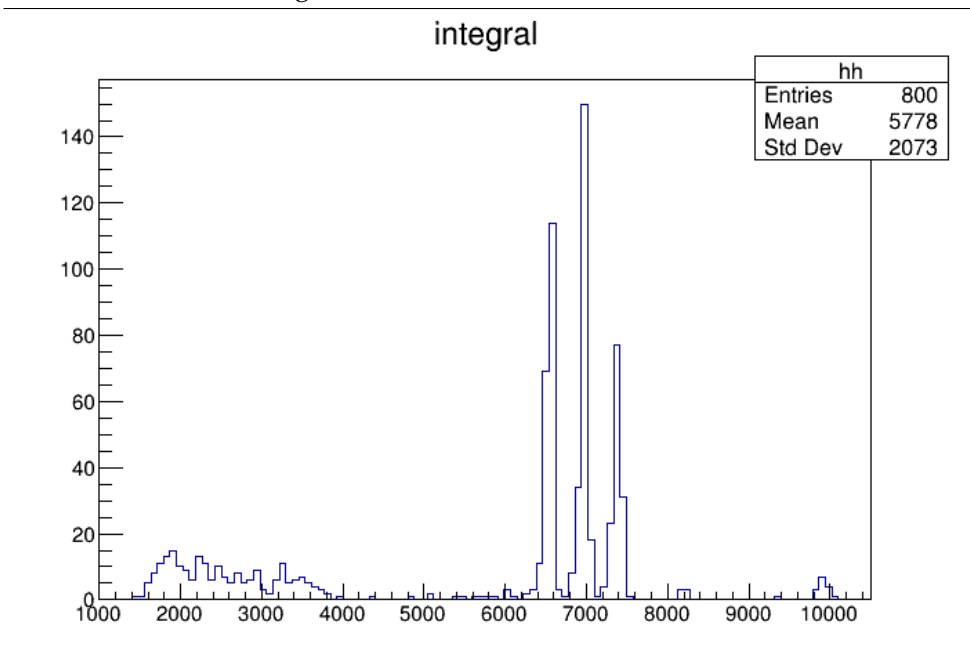


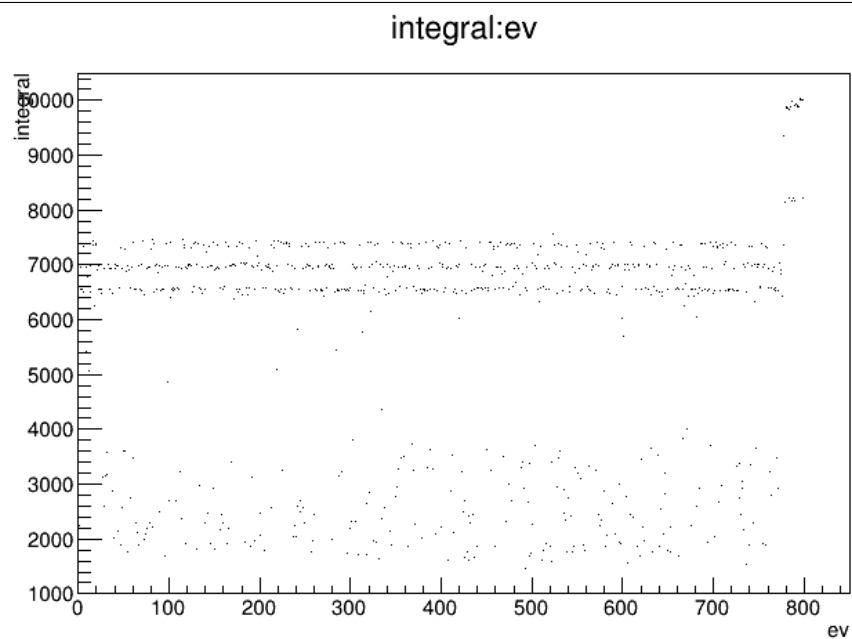
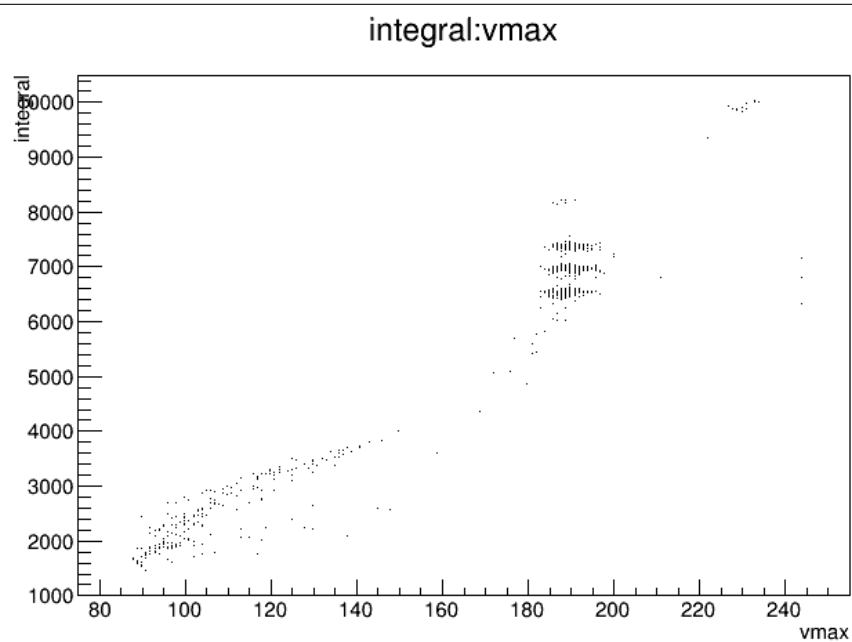
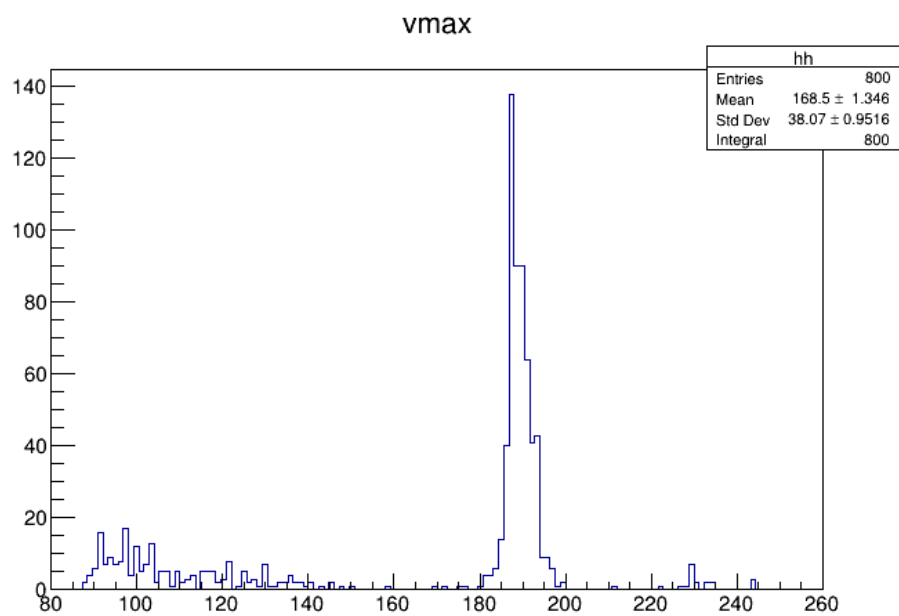
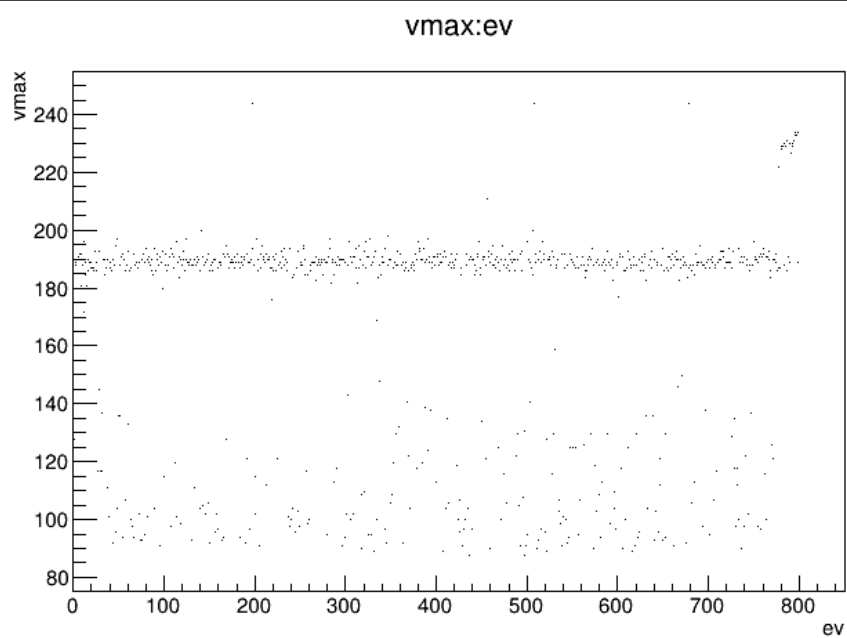
Grafico 13 Grafico integral:ev**Grafico 14** Grafico integral:vmax

Grafico 15 Grafico vmax**Grafico 16** Grafico vmax:ev

III.II Misure a pressione 550mb

Grafico 17 Grafico segnali a 550mb

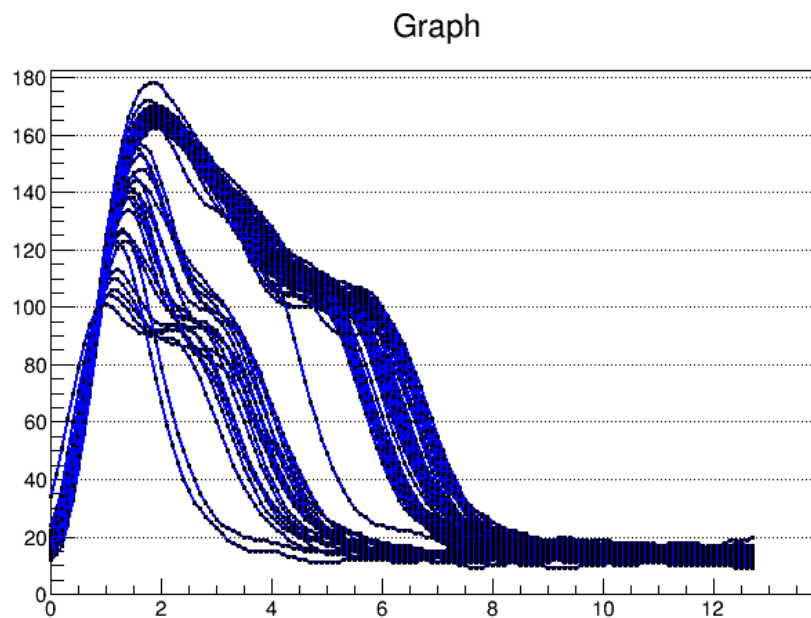


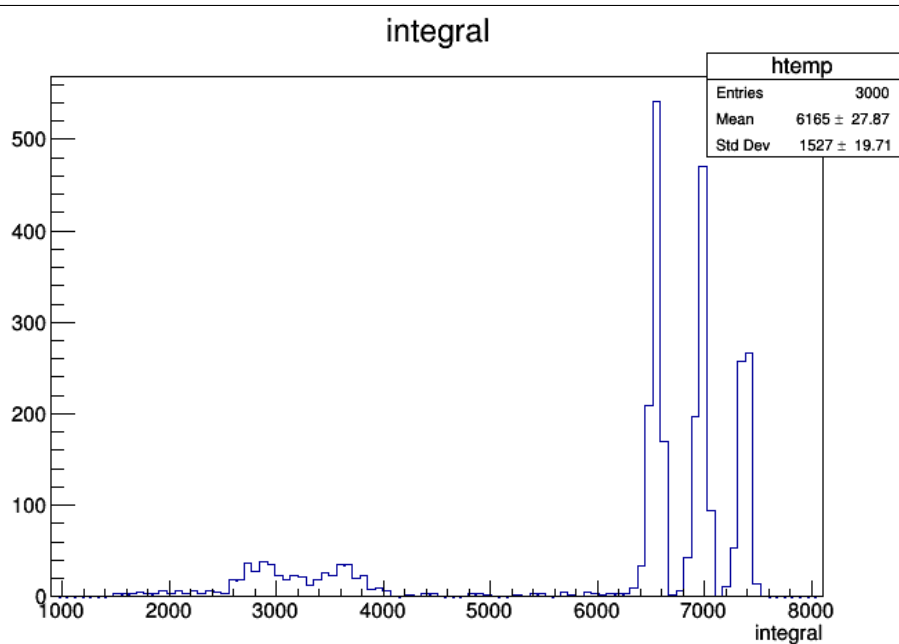
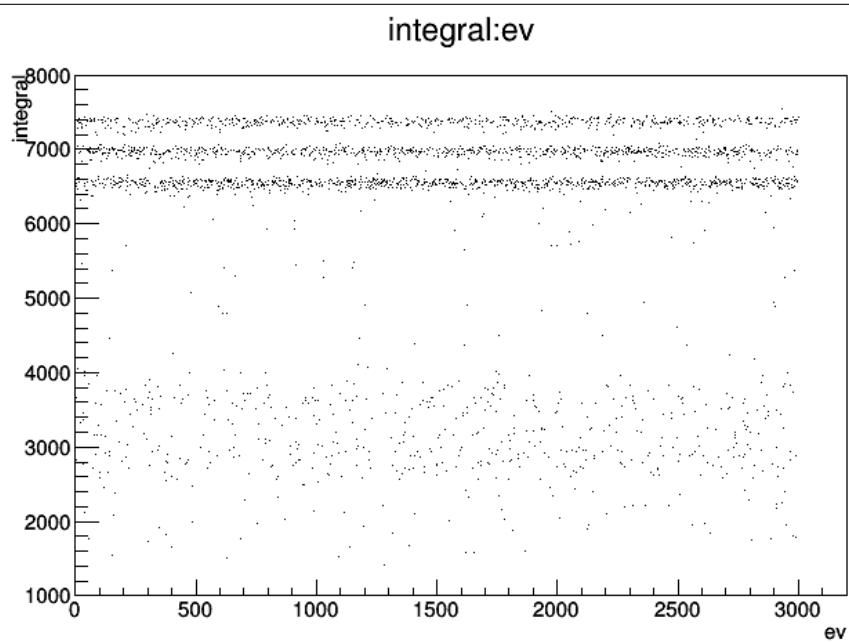
Grafico 18 Grafico integrale**Grafico 19** Grafico integral:ev

Grafico 20 Grafico integral:vmax

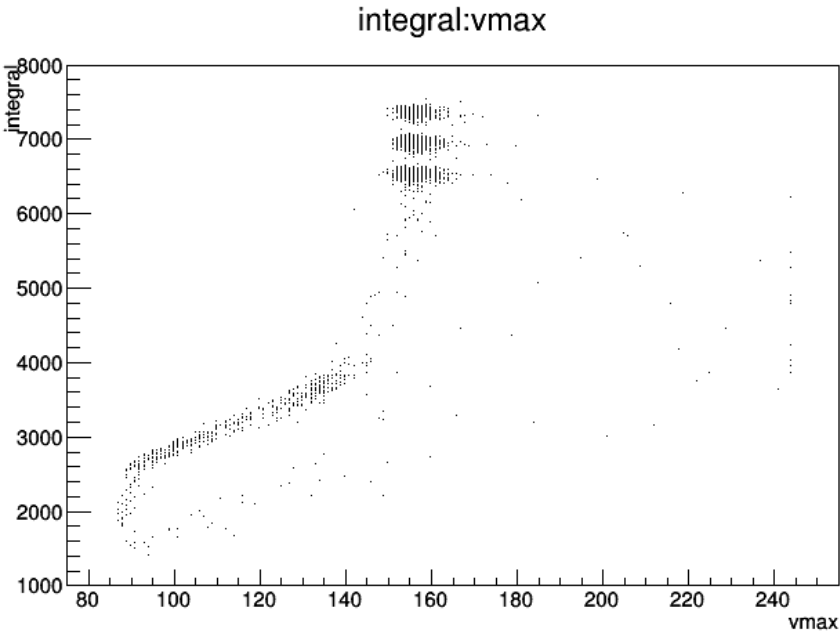


Grafico 21 Grafico vmax

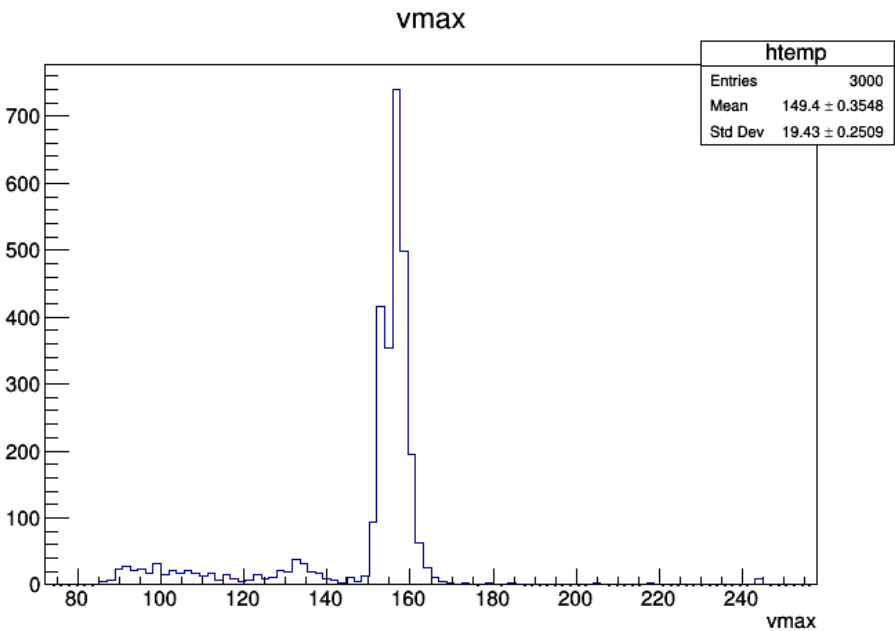
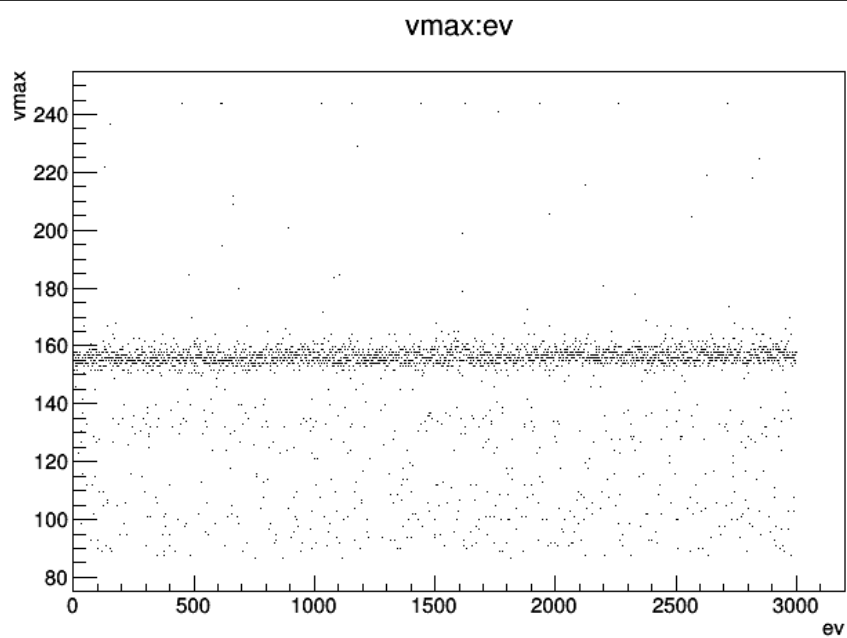


Grafico 22 Grafico vmax:ev

III.III Misure a pressione 500mb

Grafico 23 Grafico segnali a 500mb

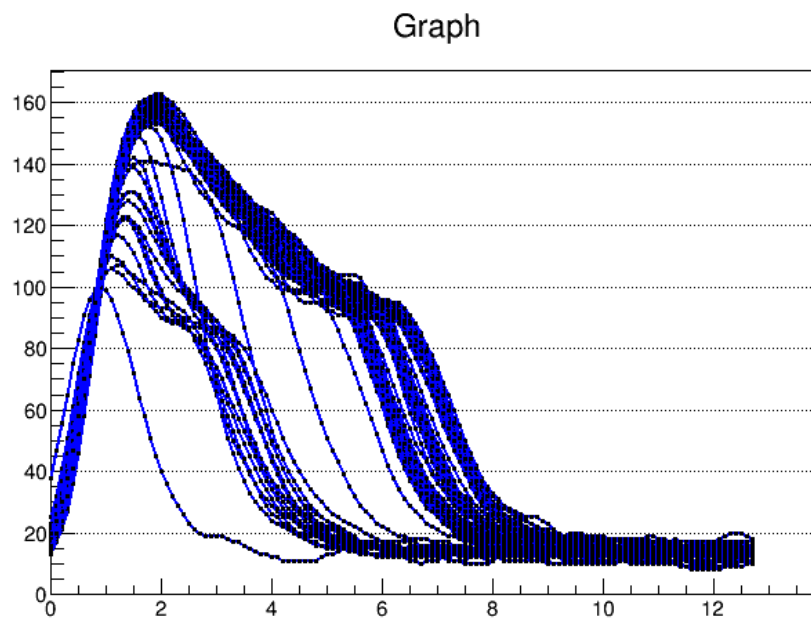


Grafico 24 Grafico integrale

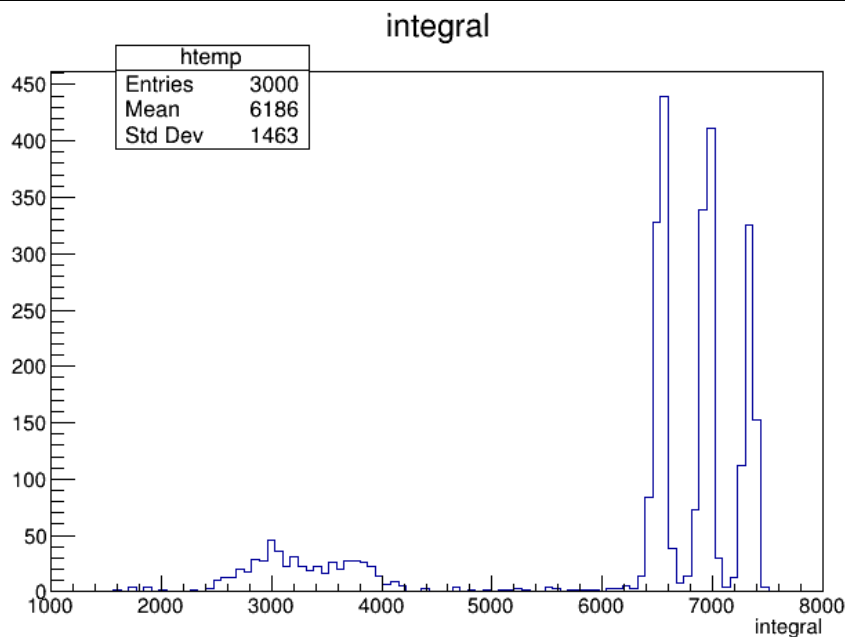


Grafico 25 Grafico integral:ev

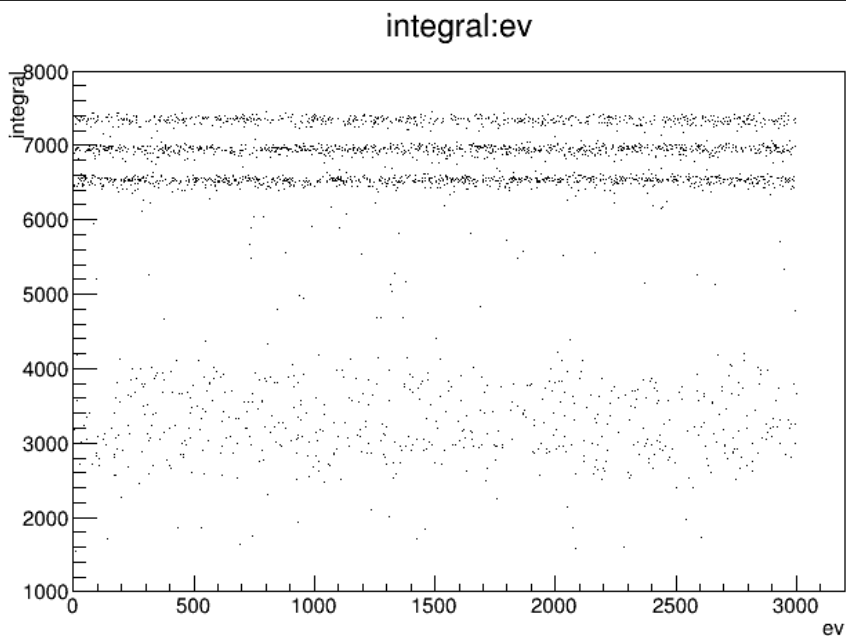


Grafico 26 Grafico integral:vmax

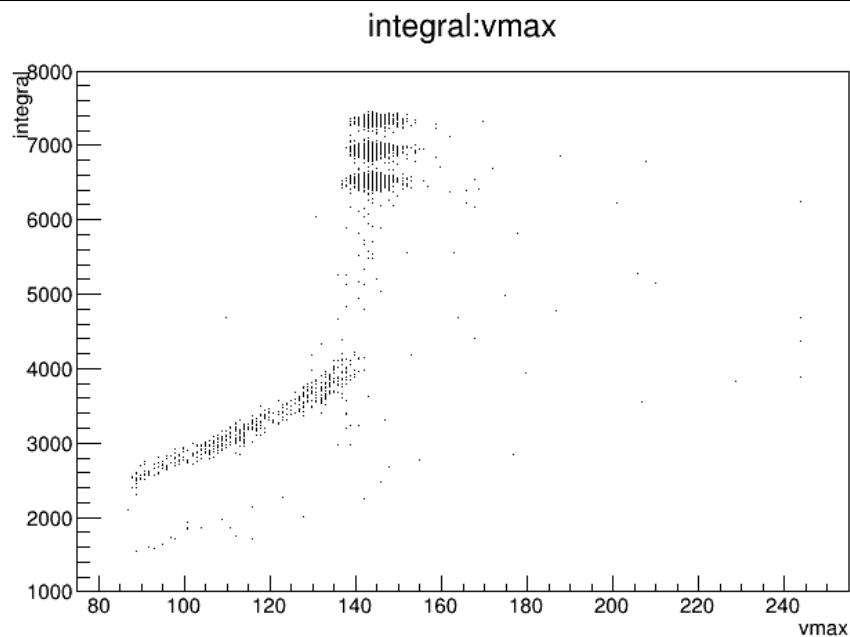


Grafico 27 Grafico vmax

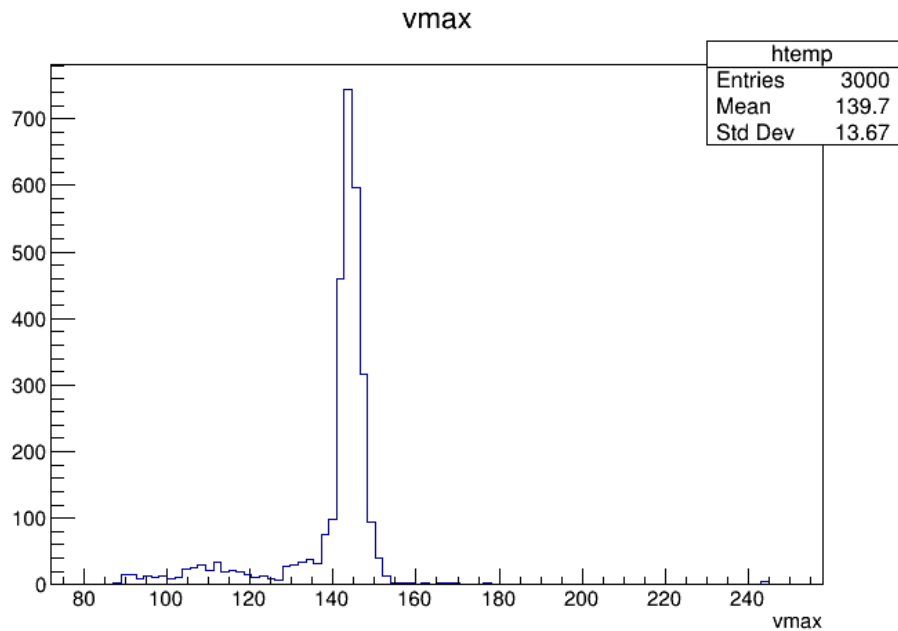
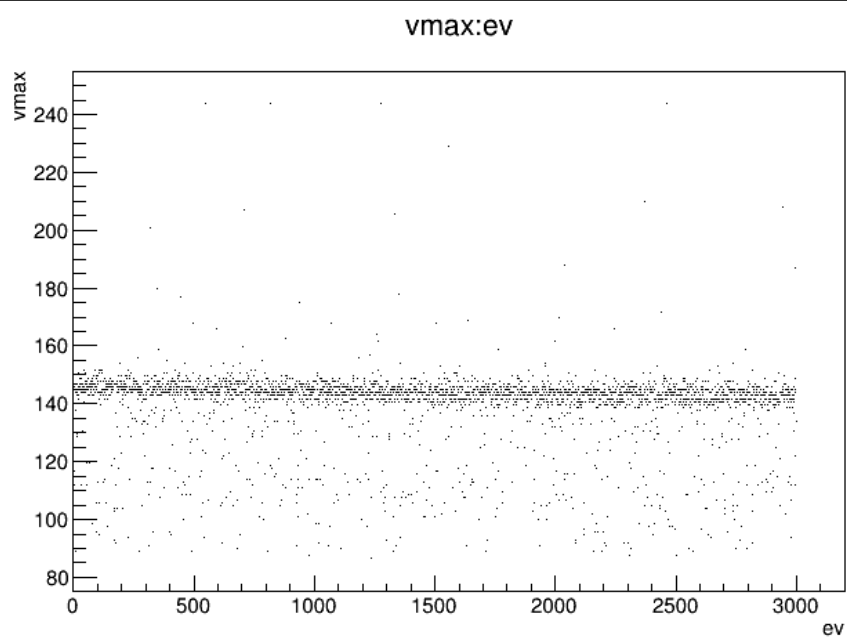


Grafico 28 Grafico vmax:ev

III.IV Misure a pressione 450mb

Grafico 29 Grafico segnali a 450mb

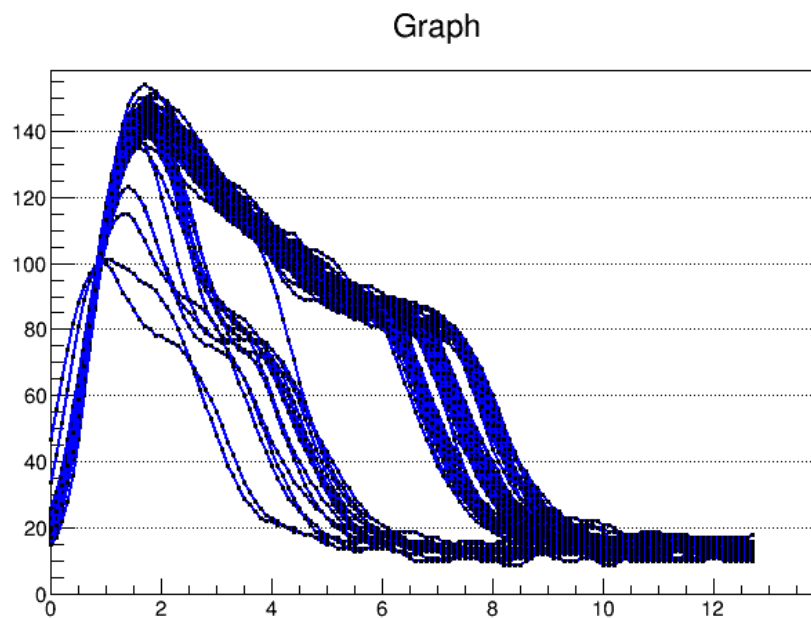


Grafico 30 Grafico integrale

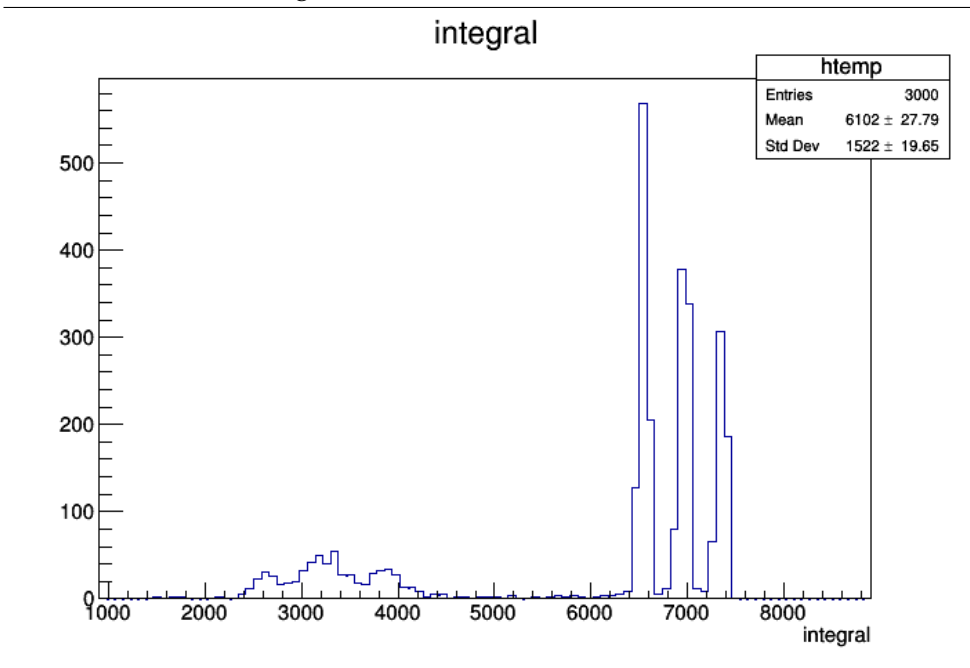


Grafico 31 Grafico integral:ev

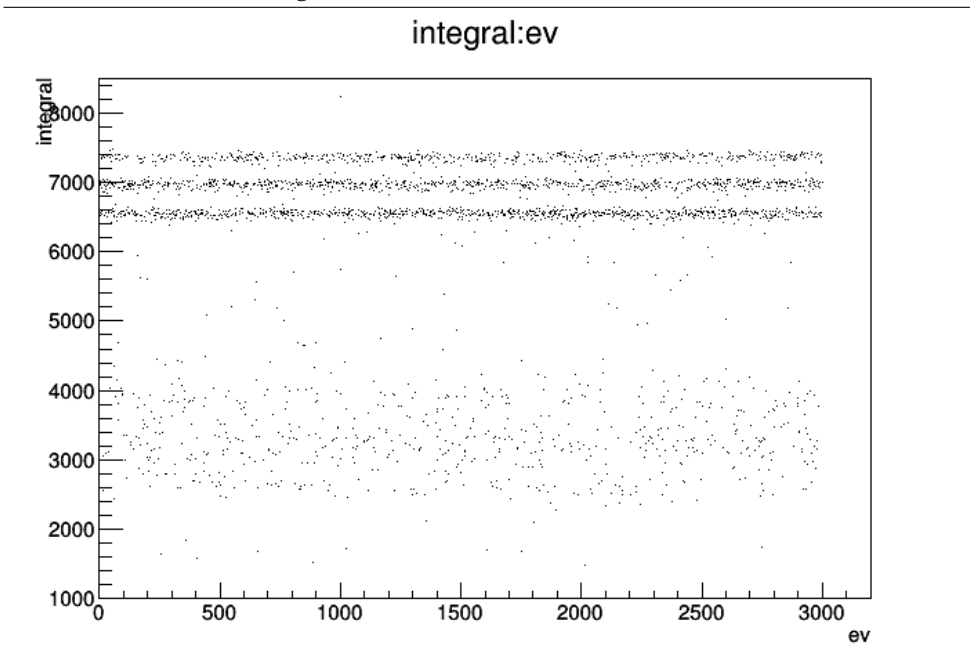


Grafico 32 Grafico integral:vmax

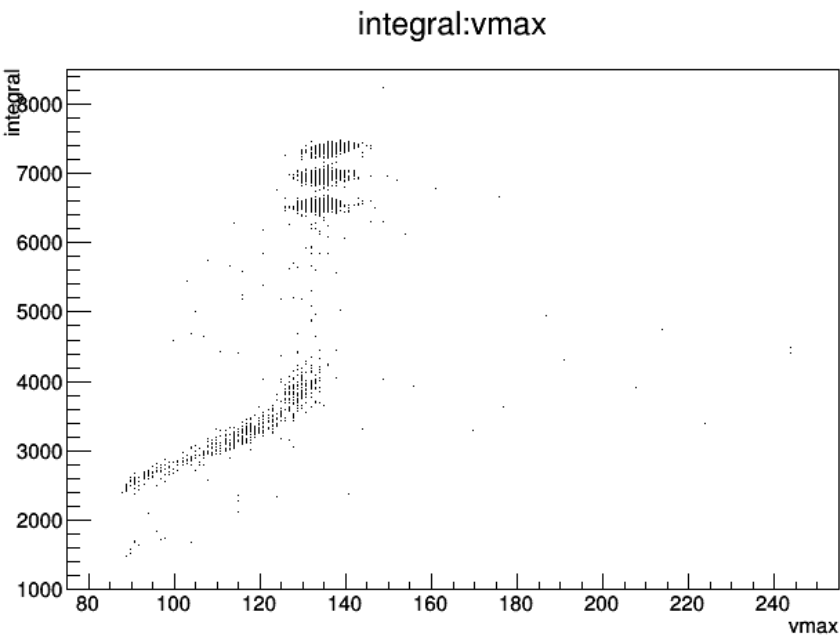


Grafico 33 Grafico vmax

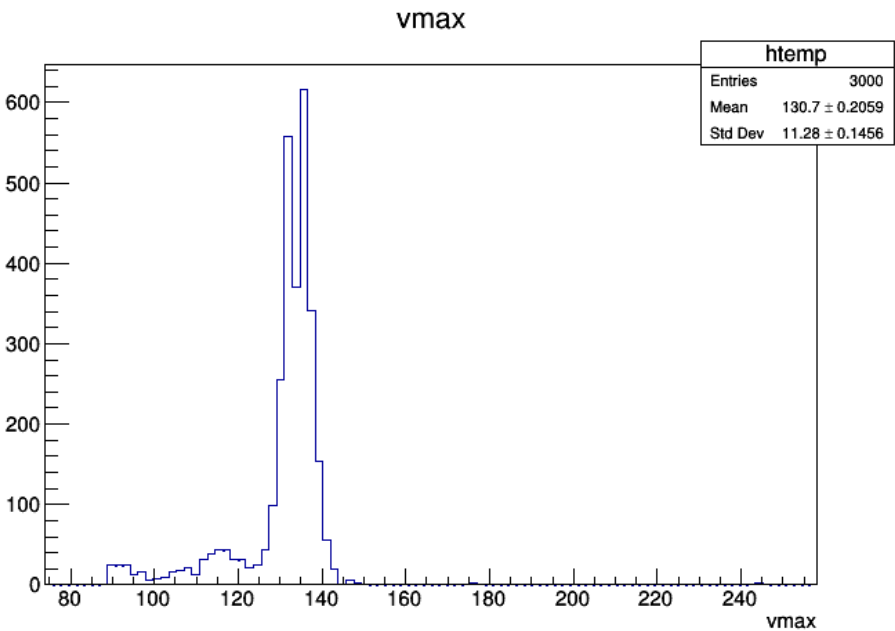
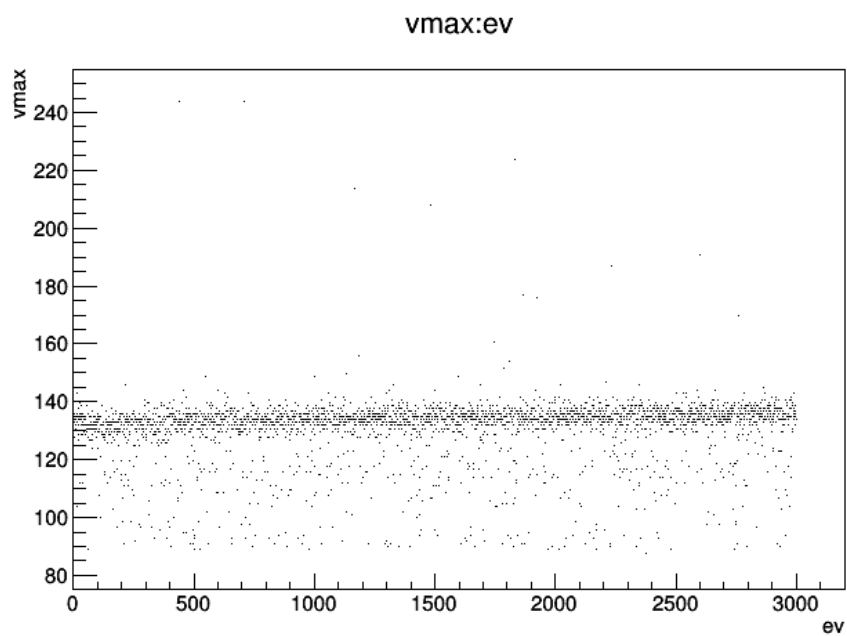


Grafico 34 Grafico vmax:ev

III.V Misure a pressione 400mb

Grafico 35 Grafico segnali a 400mb

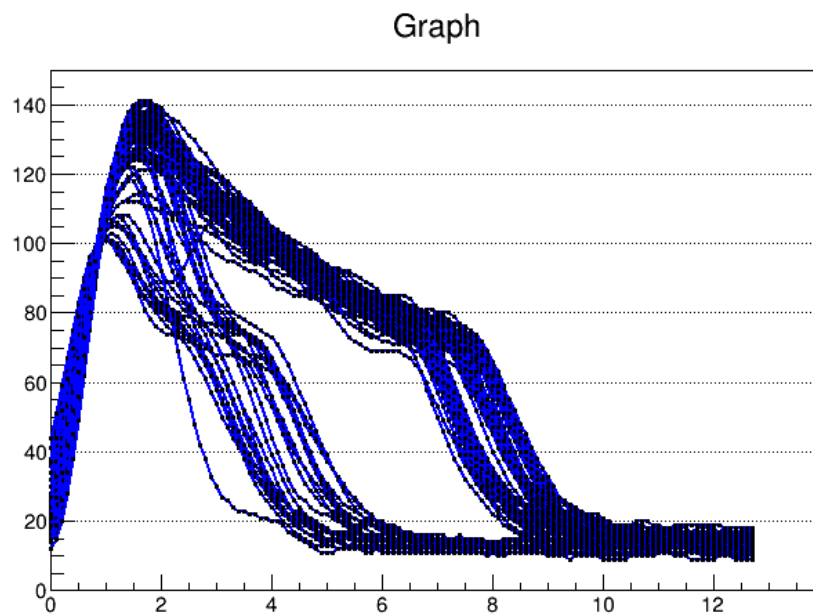


Grafico 36 Grafico integrale

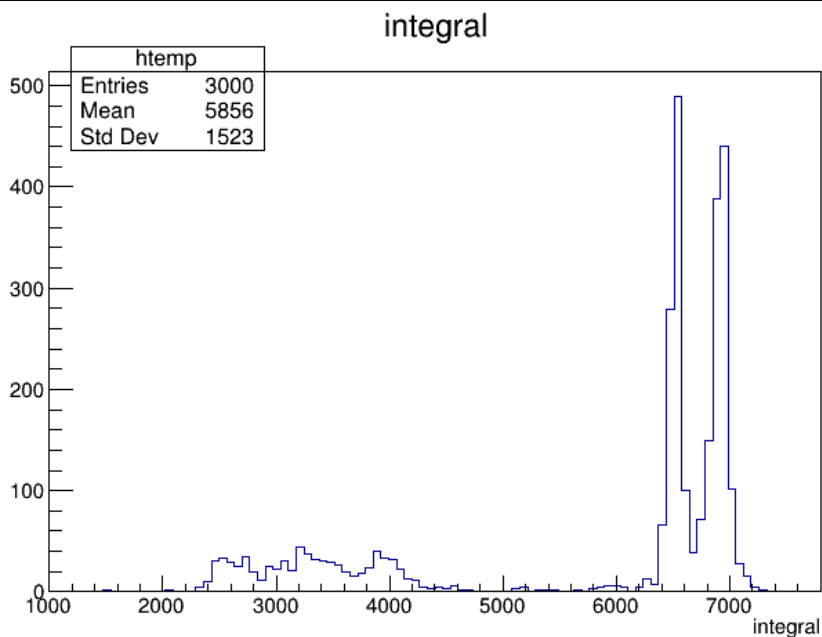


Grafico 37 Grafico integral:ev

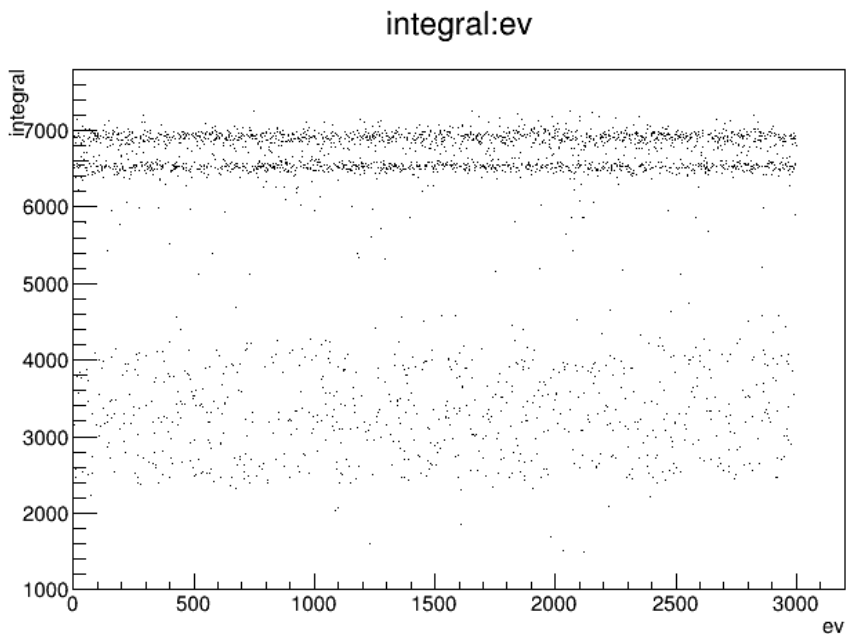


Grafico 38 Grafico integral:vmax

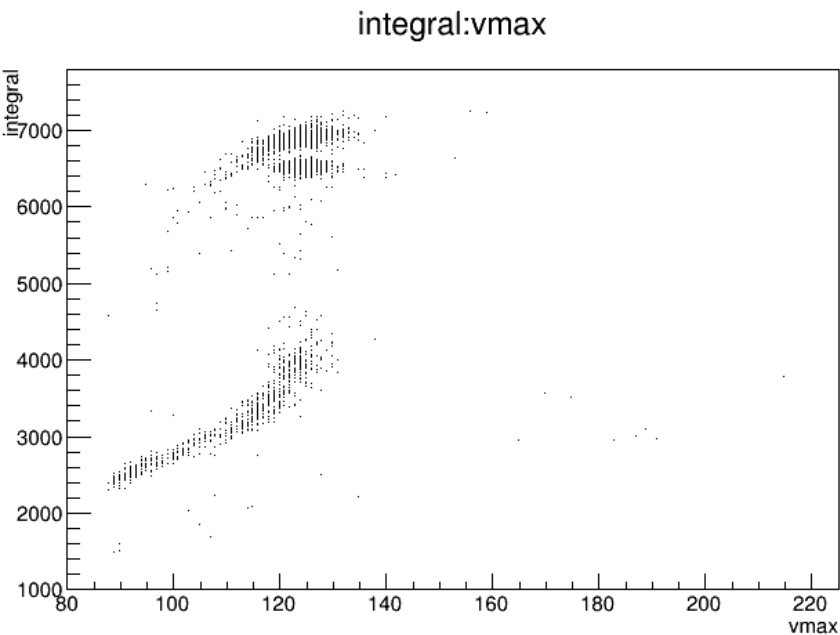


Grafico 39 Grafico vmax

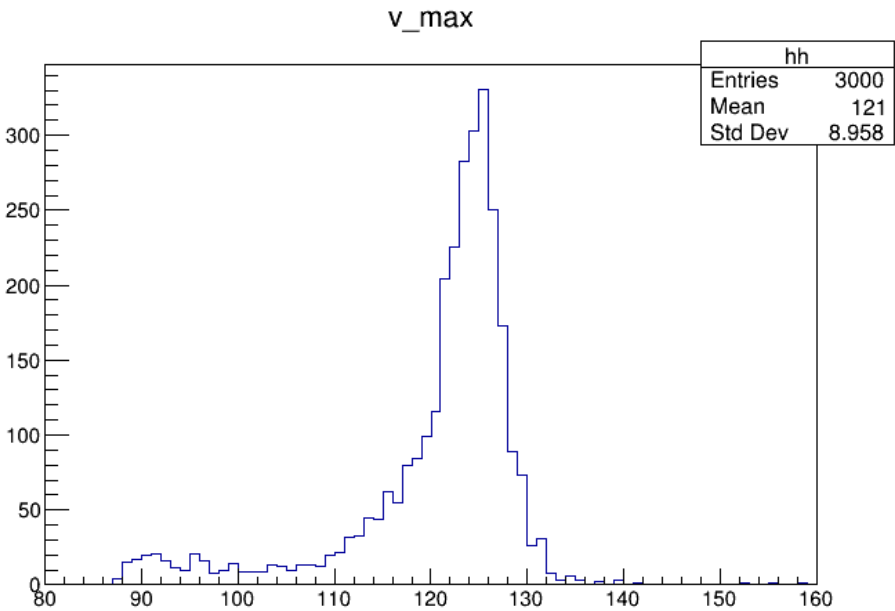
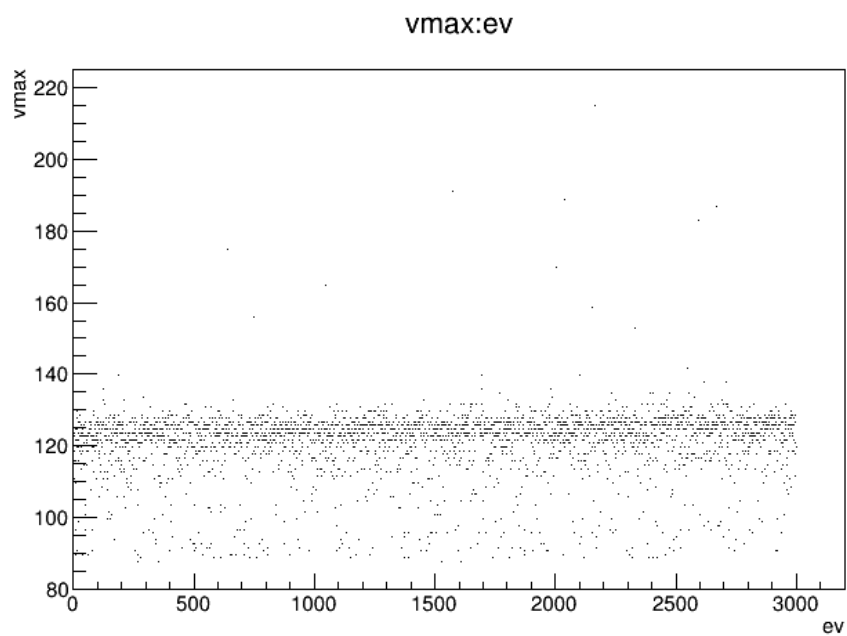


Grafico 40 Grafico vmax:ev

III.VI Misure a pressione 380mb

Grafico 41 Grafico segnali a 380mb

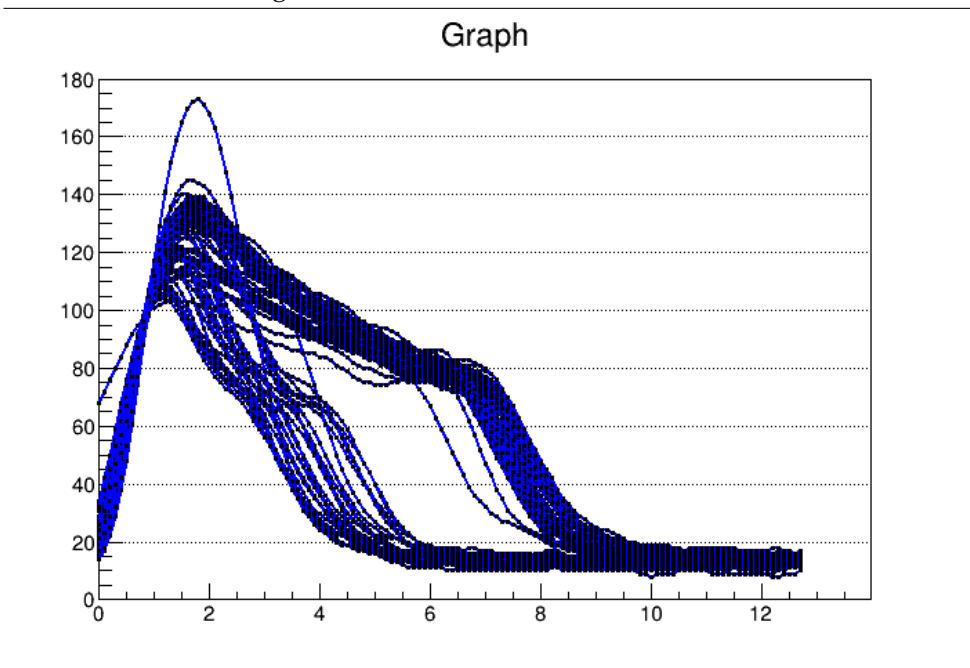


Grafico 42 Grafico integrale

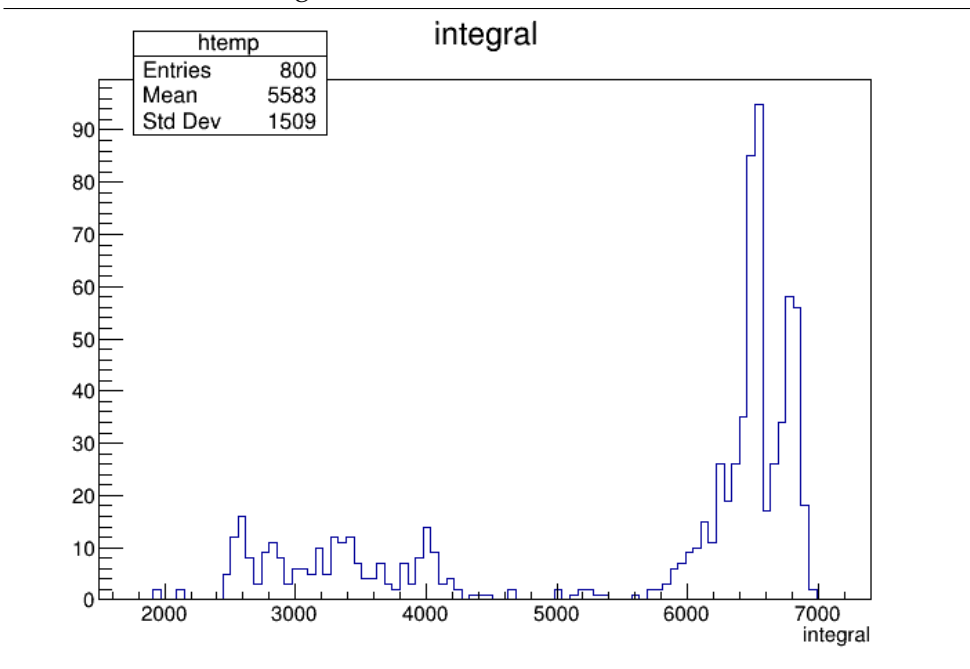


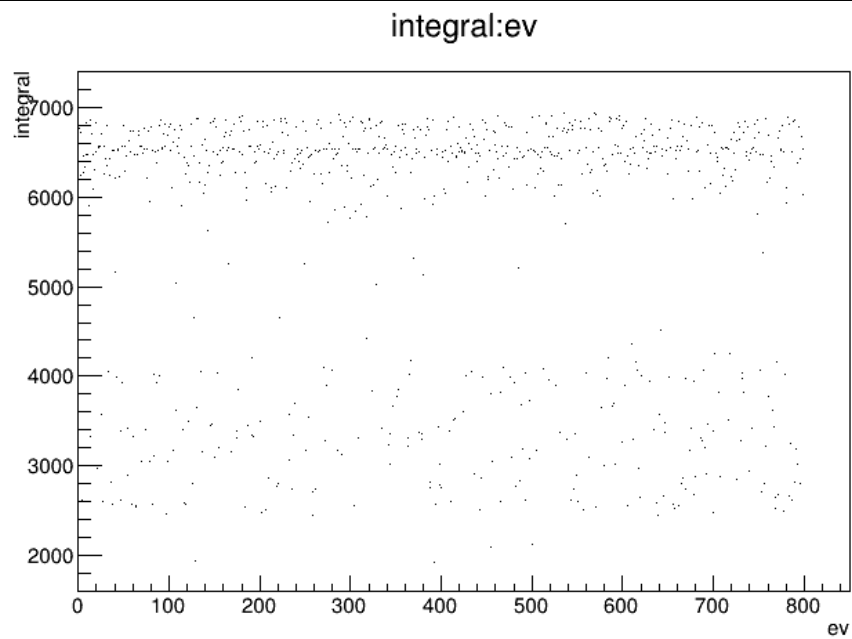
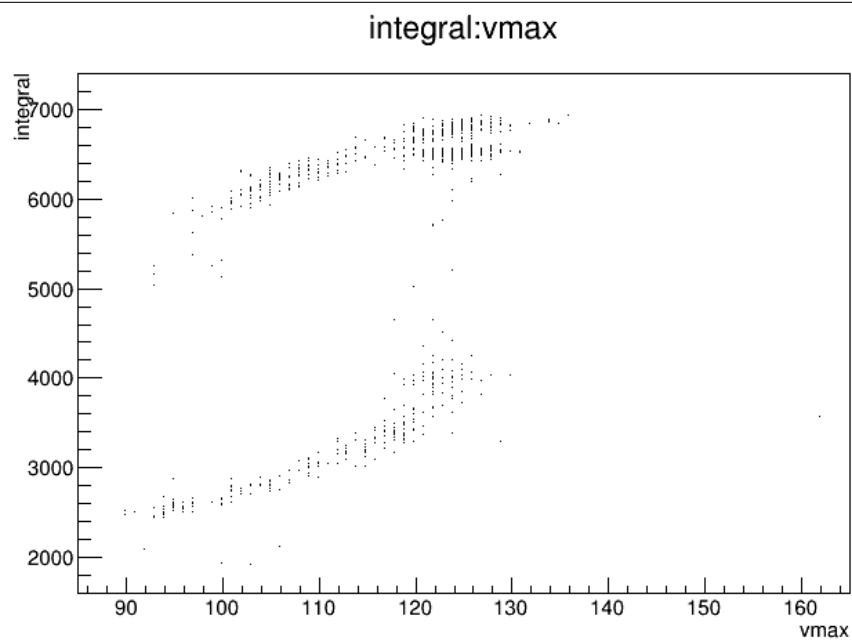
Grafico 43 Grafico integral:ev**Grafico 44** Grafico integral:vmax

Grafico 45 Grafico vmax

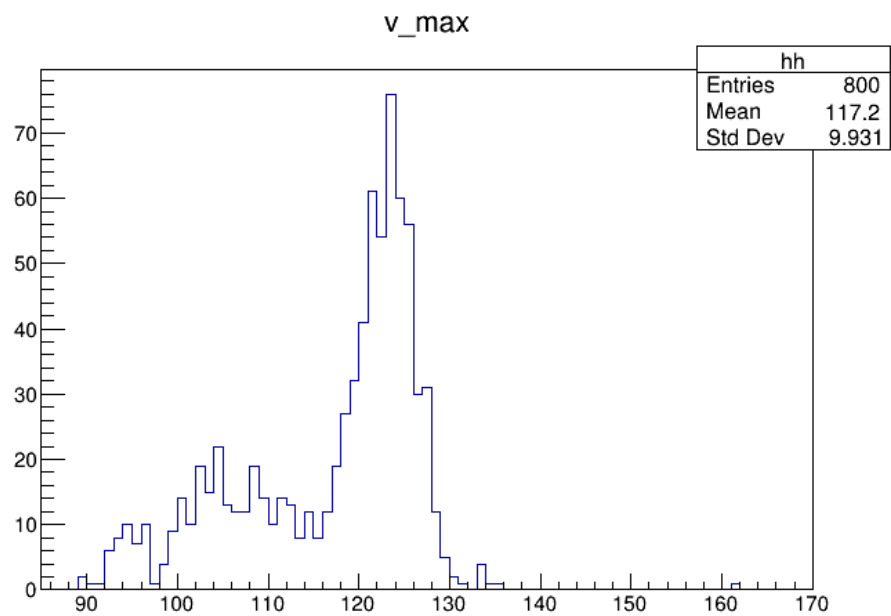
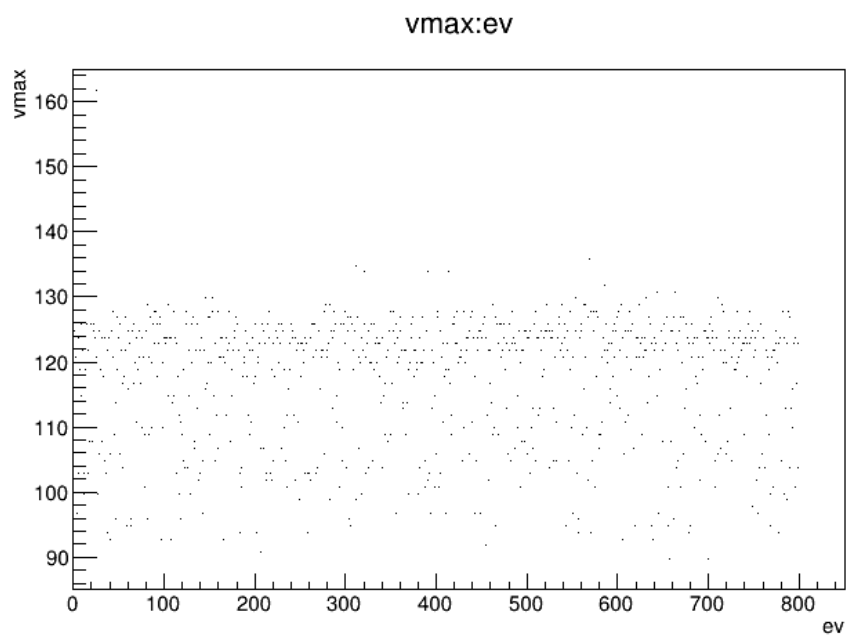


Grafico 46 Grafico vmax:ev



III.VII Curve in funzione della pressione

Sono state registrate le posizioni dei centroidi e l'RMS dei picchi in energia e dei picchi di v_{\max} delle alfa delle sorgenti.

In seguito sono stati riportati in grafico i centroidi delle energie e i centroidi dei massimi in funzione della pressione:

Come si può vedere dai grafici **Grafico 47**, **Grafico 48** e **Grafico 49**, l'integrale dei picchi, ovvero l'energia della particella alfa, rimane costante al variare della pressione, almeno finché non si arriva a pressioni troppo basse. Questo cambiamento a pressioni basse è dovuto al fatto che a bassa pressione le particelle riescono ad arrivare oltre la lunghezza della camera, e la restante carica non viene più rivelata. Addirittura, il terzo picco sparisce a basse pressioni, confondendosi con il secondo a causa dell'energia mancante.

Nel grafico **Grafico 50** invece si può chiaramente notare un andamento lineare (TODO perché?).

Tabella 1: Dati picchi in energia e RMS

p(mb)	posizione centroide	RMS
650	6552 ± 6	81 ± 4
600	6531 ± 1	44 ± 1
550	6542 ± 2	56 ± 1
500	6531 ± 2	65 ± 2
450	6547 ± 2	61 ± 1
400	6524 ± 2	62 ± 1
380	6495 ± 4	63 ± 3

Tabella 2: *Dati picchi in energia e RMS*

p(mb)	posizione centroide	RMS
650	6969 \pm 4.194	62 \pm 3
600	6951 \pm 1.737	48 \pm 1
550	6968 \pm 2.171	62 \pm 2
500	6946 \pm 1.848	55 \pm 1
450	6966 \pm 2.233	64 \pm 2
400	6903 \pm 2.473	87 \pm 2
380	6770 \pm 5.797	84 \pm 4

Tabella 3: *Dati picchi in energia e RMS*

p(mb)	posizione centroide	RMS
650	7372 \pm 5	62 \pm 4
600	7352 \pm 2	44 \pm 1
550	7370 \pm 2	56 \pm 2
500	7336 \pm 2	54 \pm 2
450	7350 \pm 3	65 \pm 2

Tabella 4: *Dati v_{max} in energia e RMS*

p(mb)	posizione centroide	RMS
650	189.3 \pm 0.1142	2.697 \pm 0.08075
600	176.8 \pm 0.0564	2.692 \pm 0.03988
550	156.4 \pm 0.0616	3.04 \pm 0.0435
500	144.2 \pm 0.05632	2.803 \pm 0.03983
450	133.2 \pm 0.0983	5.124 \pm 0.0694
400	124 \pm 0.06664	3.285 \pm 0.04712
380	122.9 \pm 0.1359	3.119 \pm 0.09609

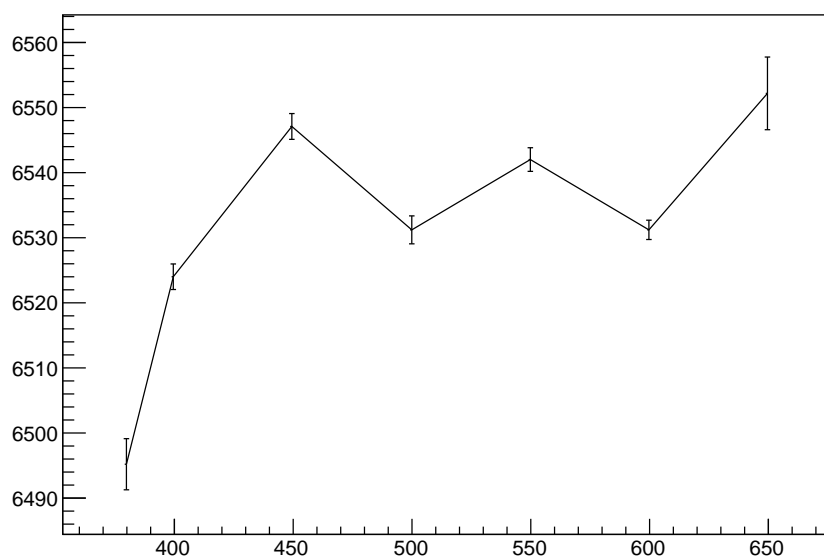
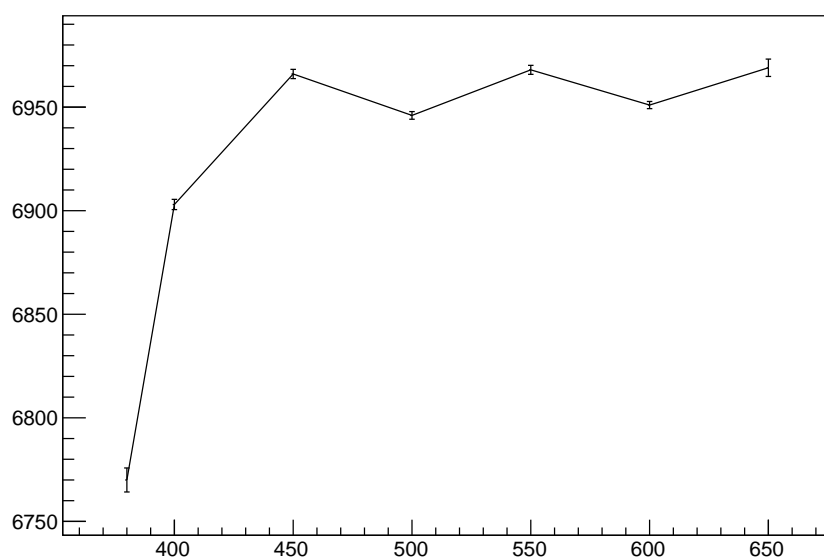
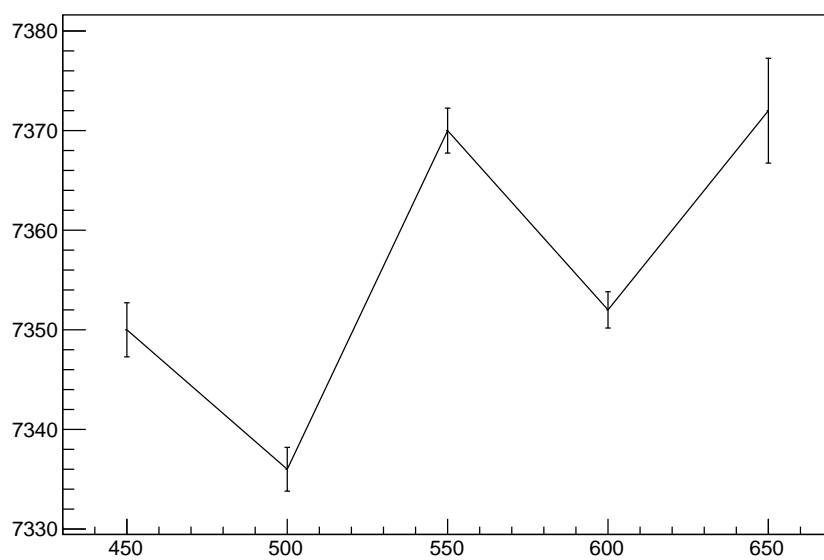
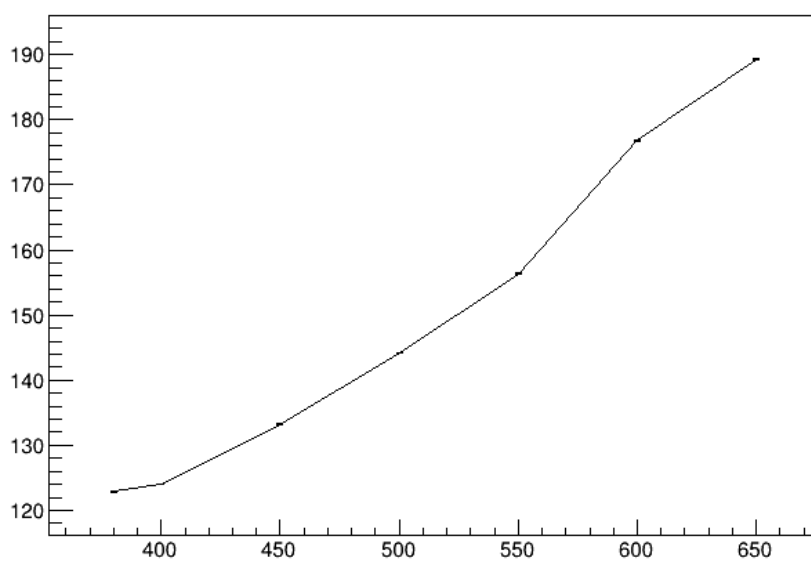
Grafico 47 Andamento integrale del picco 1 in funzione della pressione [mb]**Picchi integral 1****Grafico 48** Andamento integrale del picco 2 in funzione della pressione [mb]**Picchi integral 2**

Grafico 49 Andamento integrale del picco 3 in funzione della pressione [mb]

Picchi integral 3

**Grafico 50** Andamento integrale di v_{\max} [V] in funzione della pressione [mb]

Graph



E' stata implementata la macro *AnaBragg.C* per misurare la larghezza temporale dei segnali. Sono stati presi come riferimenti temporali i valori per i quali il segnale passa per il 40 % del valore massimo. Utilizzando la misura della larghezza temporale dei segnali a 400 mb è stata calcolata la velocità di drift, tramite la formula per il calcolo della velocità:

$$v = \frac{\text{lunghezza camera}}{\text{range temporale}}$$

La lunghezza della camera di Bragg è 120 mm.

Infine, ad alte pressioni, è stato verificato che il range spaziale fosse inversamente proporzionale alla pressione.

Tabella 5: *Tabella range spaziali e width temporali primo picco*

p(mb)	range $\pm \sigma_{\text{range}}$ (mm)	width $\pm \sigma_{\text{width}}$ (μ s)
400	113.3 \pm 0.7	69.52 \pm 0.04
450	103.2 \pm 0.6	63.33 \pm 0.04
500	94.0 \pm 0.6	57.67 \pm 0.04
550	84.6 \pm 0.5	51.93 \pm 0.03
600	72.5 \pm 0.4	44.46 \pm 0.02
650	66.9 \pm 0.4	41.06 \pm 0.05

Tabella 6: *Tabella range spaziali e width temporali secondo picco*

p(mb)	range $\pm \sigma_{\text{range}}$ (mm)	width $\pm \sigma_{\text{width}}$ (μ s)
400	124.0 ± 0.8	76.05 ± 0.06
450	111.8 ± 0.7	68.56 ± 0.05
500	102.2 ± 0.6	62.68 ± 0.05
550	91.7 ± 0.6	56.28 ± 0.03
600	78.3 ± 0.5	48.02 ± 0.03

Tabella 7: *Tabella range spaziali e width temporali terzo picco*

p(mb)	range $\pm \sigma_{\text{range}}$ (mm)	width $\pm \sigma_{\text{width}}$ (μ s)
450	119.6 ± 0.7	73.37 ± 0.05
500	109.8 ± 0.8	67.35 ± 0.05
550	98.7 ± 0.7	60.56 ± 0.05
600	84.0 ± 0.6	51.56 ± 0.04

Grafico 51 Grafico Range (mm) su Pressione (mb) picco 1

range picco 1

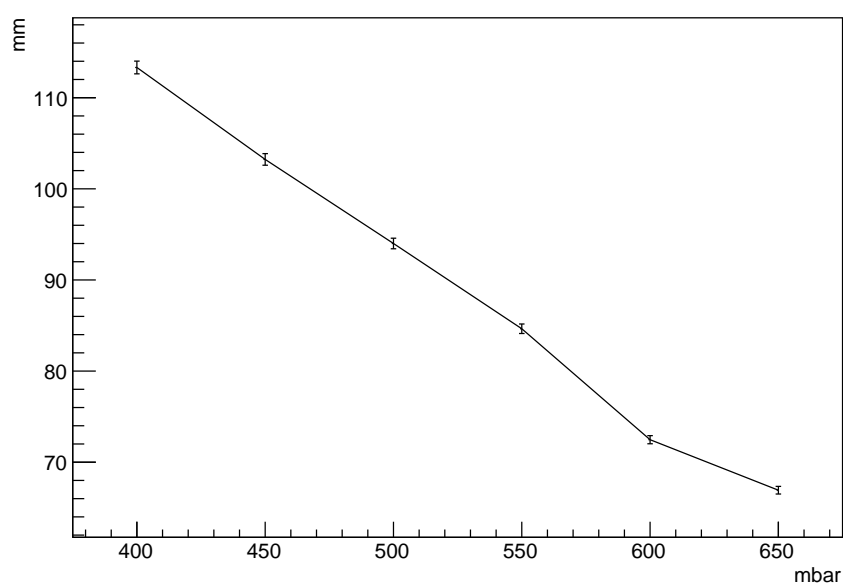
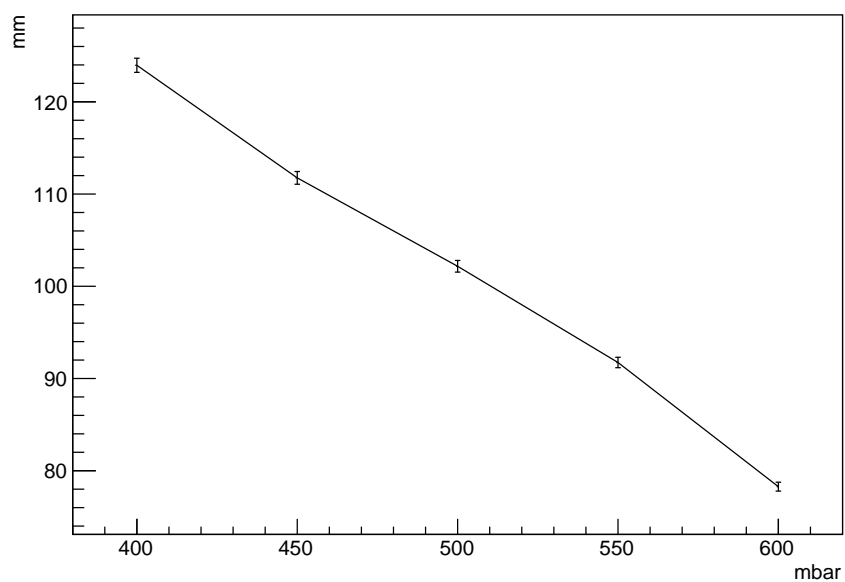


Grafico 52 Grafico Range (mm) su Pressione (mb) picco 2

range picco 2

**Grafico 53** Grafico Range (mm) su Pressione (mb) picco 3

range picco 3

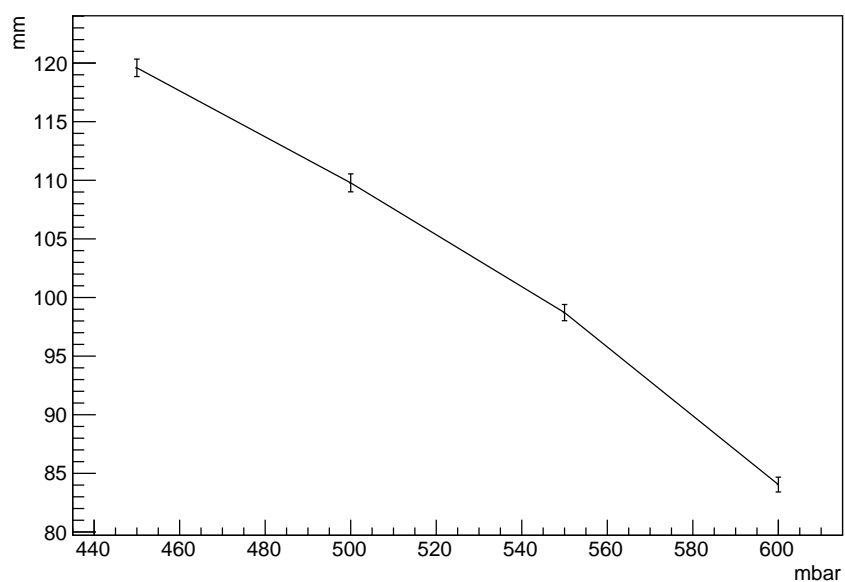


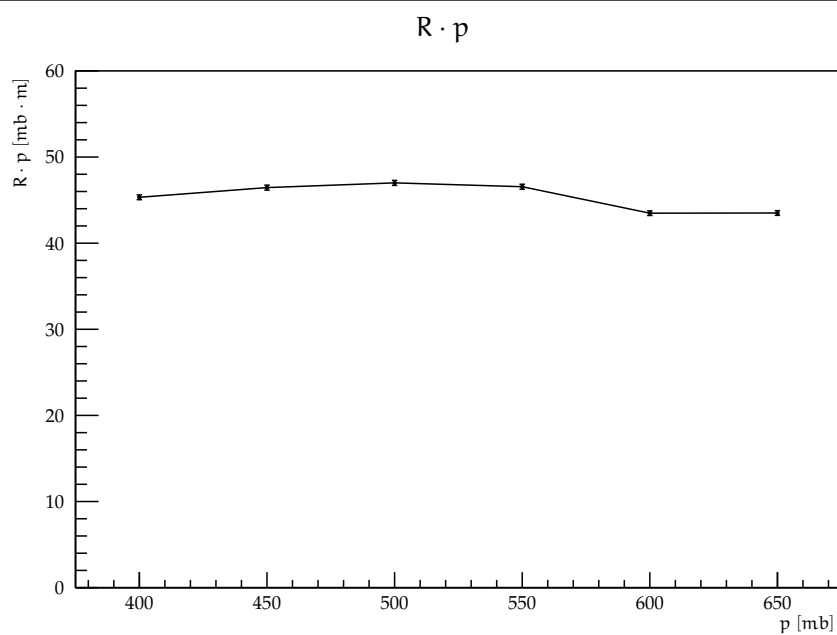
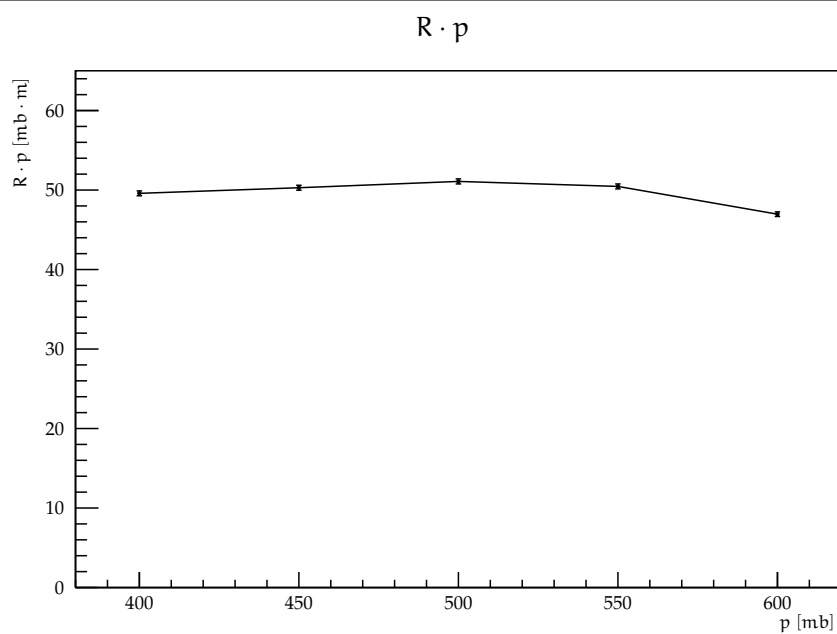
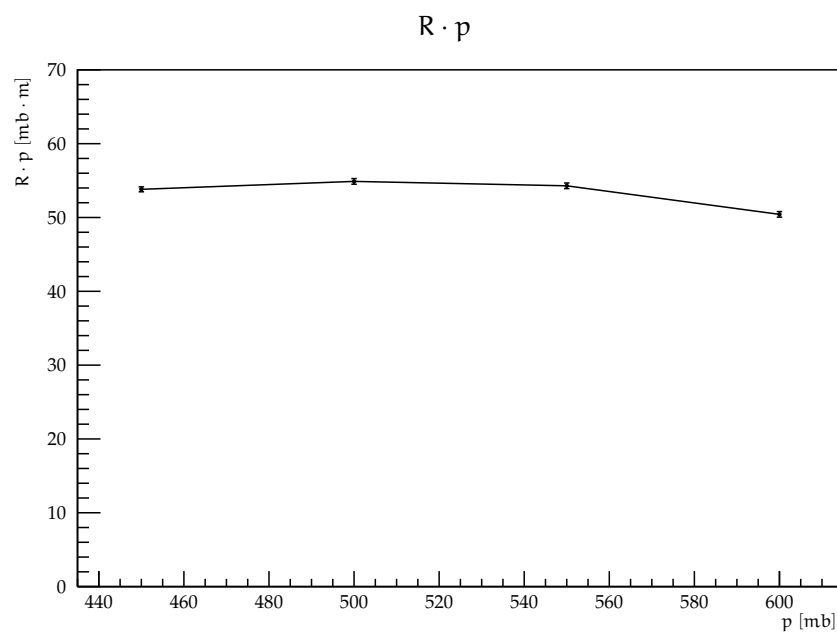
Grafico 54 Grafico Range ($m \cdot b$) su Pressione (mb)**Grafico 55** Grafico Range ($m \cdot b$) su Pressione (mb)

Grafico 56 Grafico Range ($m \cdot b$) su Pressione (mb)

IV. TABELLE

Non credo ce ne siano

Magari le tabelle dei decadimenti?

V. DISCUSSIONI E CONCLUSIONI

Da fare

VI. CODICE

È presentata qua la parte fondamentale del codice in c++ usato per i calcoli numerici.

```

1 #include <Riostream.h>
2 #include <stdlib.h>
3 #include <TR00T.h>
4 #include <TSystem.h>
5 #include "TNtuple.h"
6 #include "TFile.h"
7 #include "TTree.h"
8 #include "TCanvas.h"
9 #include "TGraph.h"
10 #include "TGraphErrors.h"
11 #include "TF1.h"
12
13 struct bragg_signal {
14     short int s[128];
15 };
16
17 int AnaBraggWidth(char *filename, int intto=80, float blfix=13, int
    nsig=0) {
18
19     bragg_signal signal;
20
21     TFile *fin=new TFile(filename);
22     if (!fin->IsOpen()) {
23         std::cout << "file not found! " << std::endl;
24         return -1;
25     }
26
27     TTree *tree = (TTree*)fin->Get("bragg");
28     if (!tree) {
29         std::cout << "Bragg tree not found! " << std::endl;
30         return -2;
31     }
32
33     TBranch *br = tree->GetBranch("signals");
34     if (!br) {
35         std::cout << "Signal branch not found! " << std::endl;
36         return -3;
37     }
38
39     br->SetAddress(&signal);
40     int nev = br->GetEntries();
41     std::cout << "Number of events in file : " << nev << std::endl;
42
43     // ANALIZZA EVENTO x EVENTO
44
45     // altri parametri iniziali

```

```

46  float thr_frac = 0.4; // soglia rispetto al vmax per il calcolo
    della larghezza
47  int intfrom = 0; // regione di integrazione da 0 a intto
48  if (intto>128) intto=128;
49  int blfrom = 108, blto = 128; // regione per il calcolo della
    baseline
50
51
52  float bl; // baseline evento x evento
53  float integral;
54  float vmax; // massimo relativo alla bl
55  float width; // larghezza dei segnali
56
57
58  char outfilename[200];
59  strcpy(outfilename,"anabragg_");
60  char *cc=strrchr(filename,'/');
61  if (cc) {cc++; strcat(outfilename,cc);}
62  else strcat(outfilename,filename);
63
64  TFile *fout=new TFile(outfilename,"RECREATE"); // output file
65
66  TNtuple *nt=new
    TNtuple("nt","", "ev:vmax:width:integral:baseline");
67
68  int maxev=nev;
69  if (nsig && nsig<nev) maxev=nsig;
70
71  // LOOP SUGLI EVENTI
72  for (int i=0; i<maxev; i++) {
73
74      // recupera l'evento
75      br->GetEntry(i);
76
77      // inizializza a zero
78      bl=0;
79      integral=0;
80      vmax=0;
81      width=0;
82
83      // calcolo baseline
84      for (int j=blfrom; j<blto; j++)
85          bl += signal.s[j]; bl /= (blto-blfrom);
86
87      // calcolo integrali e vmax
88      for (int j=intfrom; j<intto; j++) {
89          integral += (signal.s[j] - blfix);
90          if ( (signal.s[j] - blfix) > vmax ) vmax = (signal.s[j] -
            blfix);
91      }
92
93      // CALCOLO DELLA LARGHEZZA DEL SEGNALE A UNA CERTA PERCENTUALE

```

```

    DEL VMAX
94    // ...
95    float ratio = 0.4;
96    for (int j = 0; j < 128; ++j)
97    {
98        if (signal.s[j] > ratio*vmax) ++width;
99    }
100    /* alternativamente
101    int start = 0;
102    int end = 0;
103    for (int j = 0; j < 128; ++j){
104        if (!start && signal.s[j] > ratio*vmax) ++width;
105        if (start && !end && signal.s[j] > ratio*vmax) ++width;
106    }*/
107
108
109    nt->Fill(i,vmax,width,integral,bl);
110 }
111 std::cout << maxev << " events analyzed..." << std::endl;
112
113 fout->Write();
114 fout->Close();
115
116 fin->Close();
117
118 new TFile(outfilename); // riapre il file dei risultati
119
120 return 0;
121 }

```

../src/AnaBraggWidth.C

```

1 #include <Riostream.h>
2 #include <stdlib.h>
3 #include <TR00T.h>
4 #include <TSystem.h>
5 #include "TNtuple.h"
6 #include "TFile.h"
7 #include "TTree.h"
8 #include "TCanvas.h"
9 #include "TGraph.h"
10 #include "TGraphErrors.h"
11 #include "TF1.h"
12
13 struct bragg_signal {
14     short int s[128];
15 };
16
17 int plotSignal(bragg_signal sig, int same) {
18
19     float x[128]; for (int i=0; i<128; i++) x[i]=i*0.1;
20     float y[128]; for (int i=0; i<128; i++) y[i]=sig.s[i];

```

```

21 TGraph *g = new TGraph(128,x,y); // crea il grafico
22 g->SetMarkerStyle(7); // imposta alcuni attributi
23 g->SetLineColor(4);
24 g->SetLineWidth(2);
25
26 TCanvas *csig = (TCanvas*)gROOT->FindObject("csig"); // cerca
    l'oggetto "csig" (canvas)
27 if (!csig) {
28     csig=new TCanvas("csig"); // se non c'e' la crea nuova
29     csig->SetGridy();
30     g->Draw("apl"); // disegna il grafico e anche il frame con gli
        assi
31 }
32 else {
33     csig->cd(); // se c'e' si posiziona sulla canvas "csig"
34     if (same)
35         g->Draw("pl"); // disegna nel frame esistente
36     else {
37         csig->Clear();
38         g->Draw("apl"); // disegna in un nuovo frame
39         gSystem->Sleep(200); // aspetta 200 ms
40     }
41 }
42 csig->Modified(); // aggiorna la canvas
43 csig->Update();
44 gSystem->ProcessEvents(); // aggiorna la grafica
45
46 return 0;
47 }
48
49 int PlotSignals(char *filename, int plfrom=0, int plto=100, int
    same=1) {
50
51     bragg_signal signal;
52
53     TFile *fin=new TFile(filename);
54     if (!fin->IsOpen()) {
55         std::cout << "file not found! " << std::endl;
56         return -1;
57     }
58
59     TTree *tree = (TTree*)fin->Get("bragg");
60     if (!tree) {
61         std::cout << "Bragg tree not found! " << std::endl;
62         return -2;
63     }
64
65     TBranch *br = tree->GetBranch("signals");
66     if (!br) {
67         std::cout << "Signal branch not found! " << std::endl;
68         return -3;
69     }

```

```
70 |
71 | br->SetAddress(&signal);
72 | int nev = br->GetEntries();
73 | std::cout << "Number of events in file : " << nev << std::endl;
74 |
75 | for (int i=plfrom; i<plto; i++) {
76 |     br->GetEntry(i);
77 |     plotSignal(signal,same);
78 | }
79 |
80 | return 0;
81 | }
```

../src/PlotSignals.C