



UNIVERSITA' DEGLI STUDI DI FERRARA
Dipartimento di Matematica ed Informatica
Corso di Laurea in Informatica

Sviluppo di un sistema client-server e di interfacce grafiche per l'analisi spettrometrica a bordo di un rivelatore portatile di radiazione gamma.

Relatore:

Dott. Mantovani Fabio

Laureando:

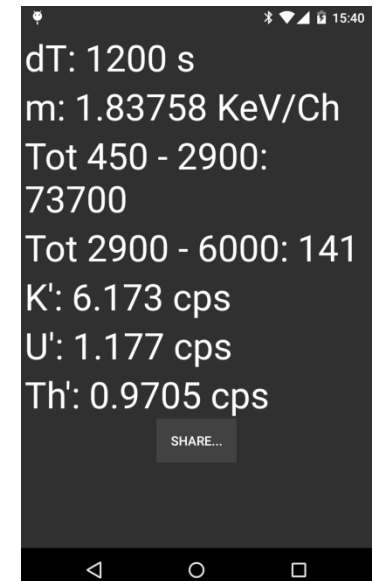
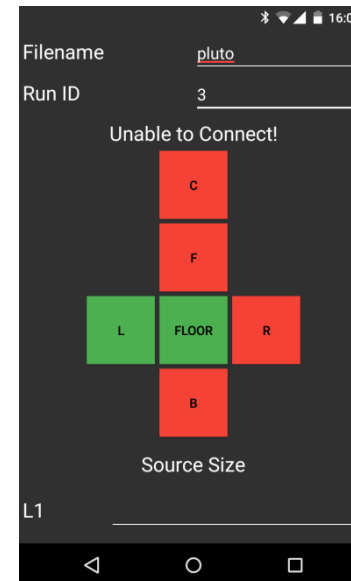
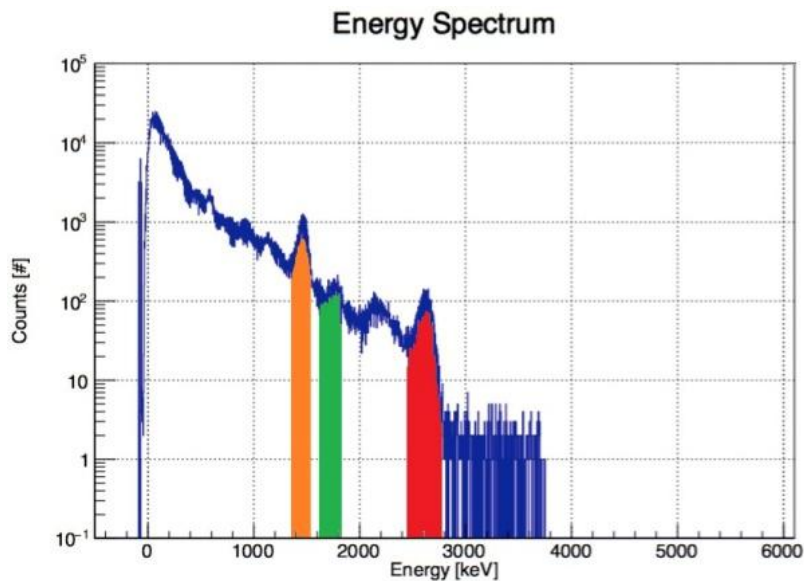
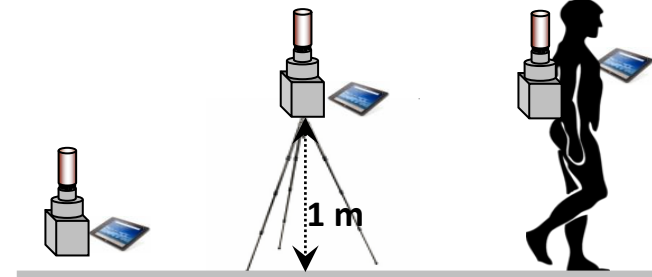
Chiarelli Enrico

Correlatore:

Dott. Turisini Matteo

Overview

- Le misure di radioattività gamma outdoor
- Il sistema rilevatore-MCA
- Punti critici dell'impiego di MCA e obiettivi
- Il programma GammaServer
- Il programma GammaClient per PC
- L'applicazione Android GammaClient
- Conclusioni



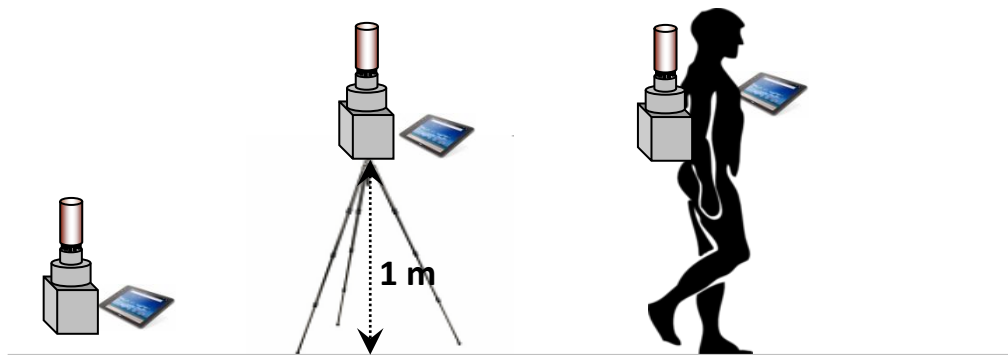
Spettroscopia Gamma Outdoor

Progetto di ricerca: misurare radionuclidi emettitori gamma outdoor



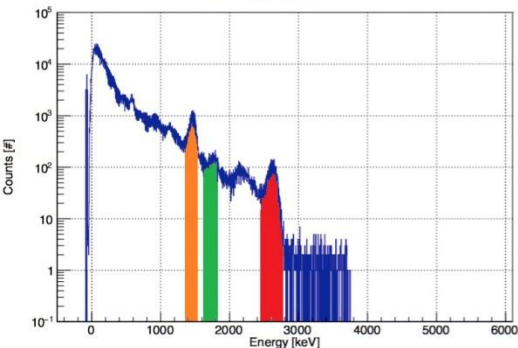
Contesti in cui si svolgono le misure:

- Miniera
- Discarica
- Siti inquinati
- Airborne (UAV)



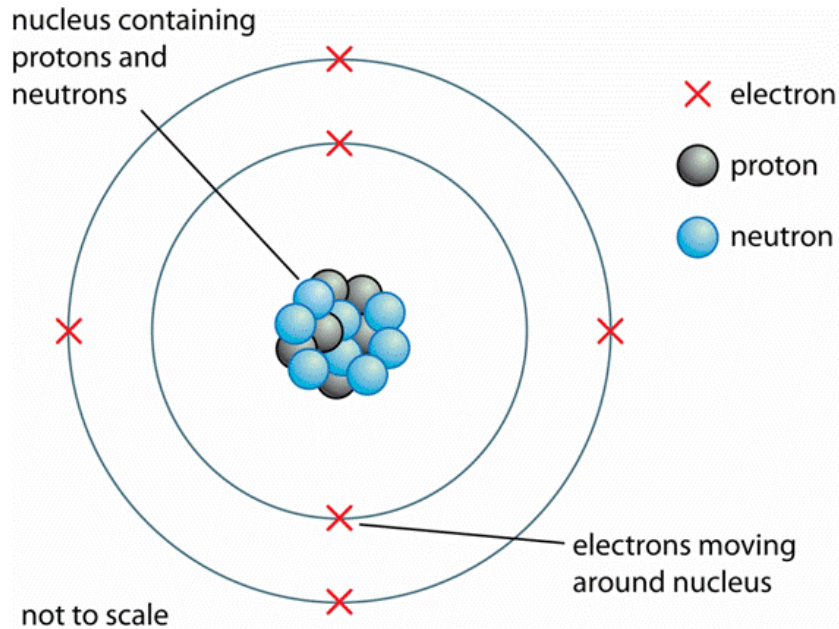
Misurazione abbondanze **Potassio**, **Uranio** e **Torio**

Energy Spectrum

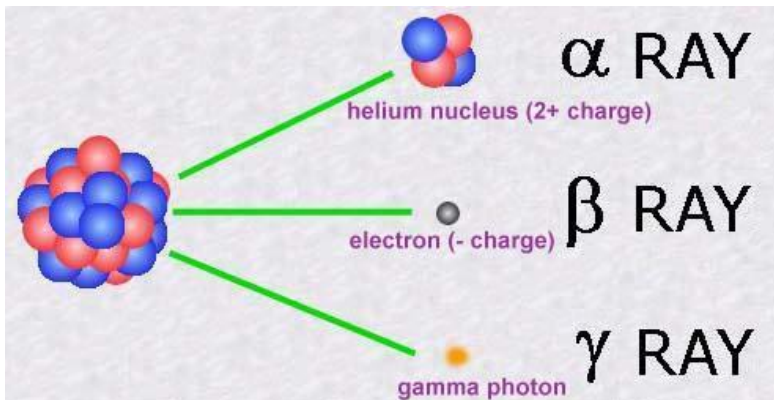


Radioisotope	Daughter	Energy (keV)	Half life	Typical abund.
^{40}K	/	1460	1.3 Gy	2%
^{238}U	^{214}Bi	1765	4.5 Gy	3 ppm
^{232}Th	^{208}Tl	2614	14.1 Gy	10 ppm

Cenni di radioattività



- Gli atomi sono formati da Protoni, Neutroni ed Elettroni.
- Non tutte le configurazioni dei nuclei degli atomi sono stabili.
- Le configurazioni instabili dei nuclei tendono a **decadere** nel tempo, **cedendo massa e/o energia** all'ambiente circostante.



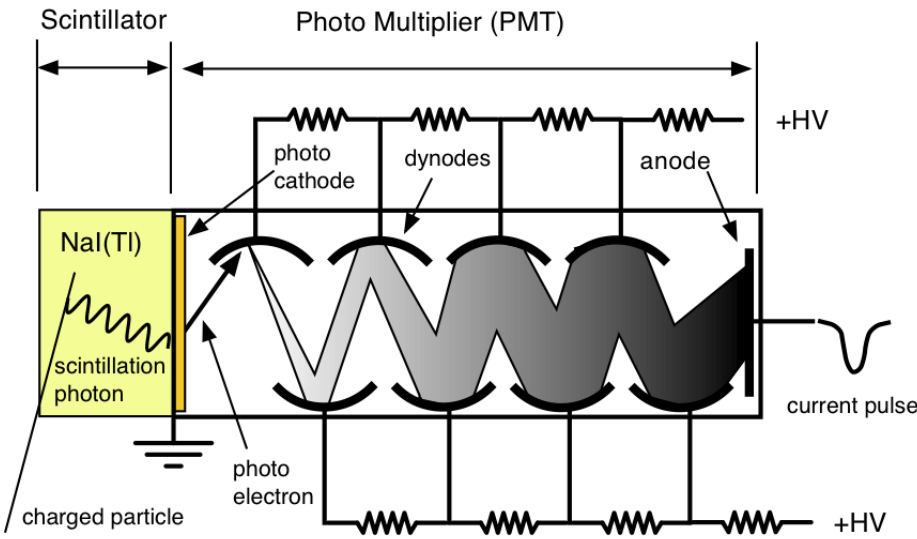
- I decadimenti radioattivi sono di 3 tipi:
- La radiazione α : nuclei di elio.
 - La radiazione β : elettroni e neutrini
 - La radiazione γ : fotoni.

I rilevatori NaI(Tl)

- I rilevatori NaI(Tl) vengono attivati dall'interazione con i raggi γ .

- I raggi γ producono una luce di scintillazione all'interno del cristallo del rivelatore.

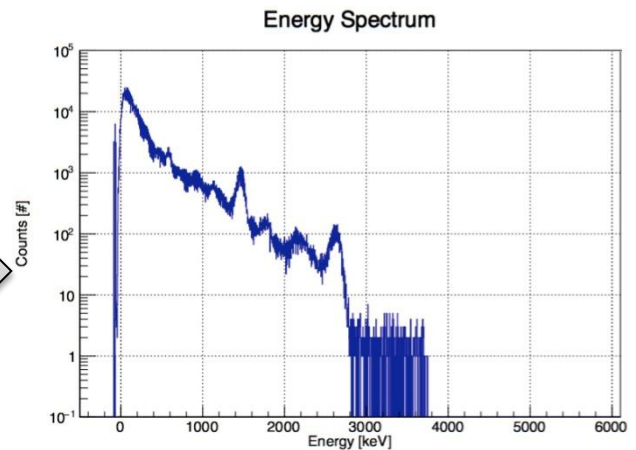
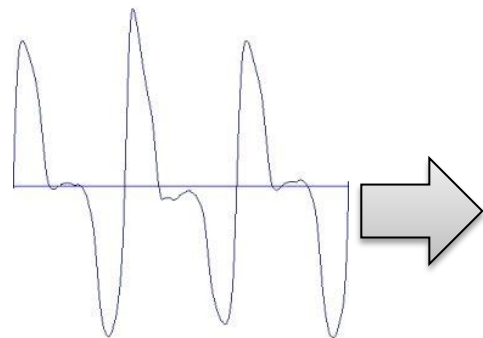
- La luce di scintillazione viene trasformata in un debole segnale elettrico dal fotocatodo ed amplificata dal PMT.



Raggio γ (E)



Segnale elettrico

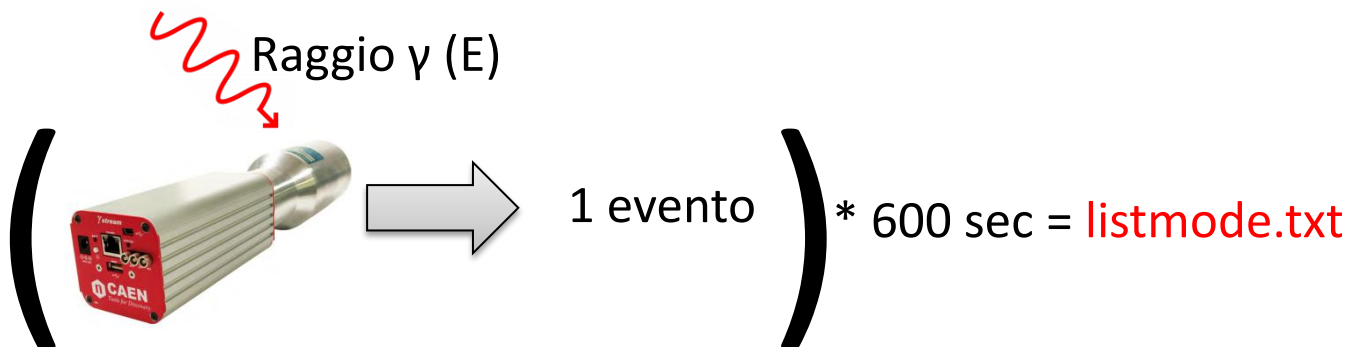


Il Multi Channel Analyzer γ Stream



γ Stream è un Multi Channel Analyzer (MCA) che si interfaccia con i rilevatori a scintillazione NaI(Tl).

Il software a bordo di γ Stream traduce i segnali elettrici provenienti rilevatore in “eventi”, conservandoli su **file**.

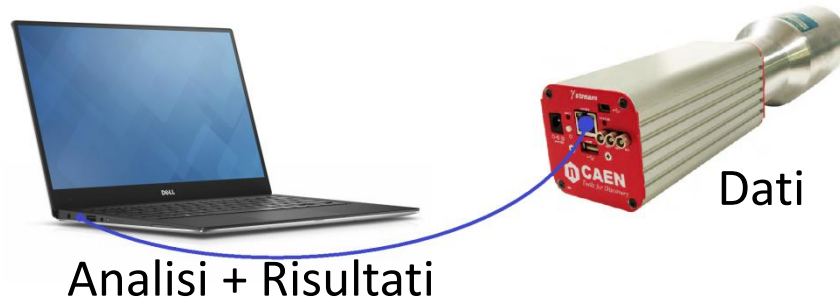


```
#GammaStream 1.0 LIST
#StartTime: 2016-02-04T13:57:27
#Fields: Time      Energy      Gain
          185896      265         1.000
          307634      172         1.000
          543628      483         1.000
          ....
337322968731      0          1.000
          ...
337435019792      42         1.000
337435153156      163        1.000
# End of RUN
```

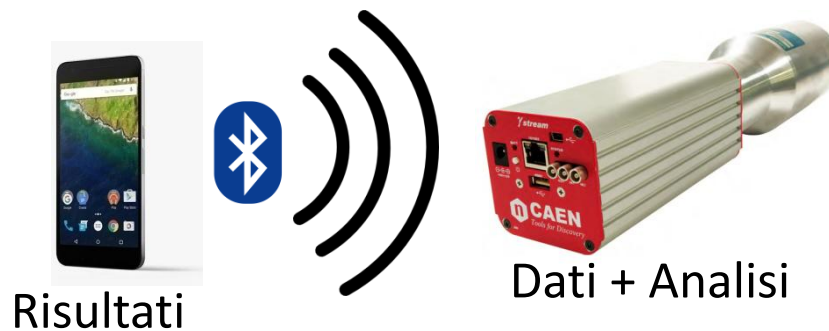
Una misura ha 600 secondi di durata e produce un **file ASCII** composto dalla lista di tutti gli eventi registrati durante questo tempo e le rispettive energie.

Punti critici e soluzioni

- Al fine di eseguire l'analisi è necessario **trasferire i dati con un cavo RJ-45**.
- Per svolgere l'analisi è necessario l'impiego di un **PC**, poco pratico nel contesto.
- Gli algoritmi di analisi vengono eseguiti da **riga di comando**, sono **privi di una UI**.



- Il **Bluetooth** elimina la necessità del cavo RJ-45 e **l'analisi viene eseguita su yStream**.
- L'utilizzo di uno **Smartphone Android** rimuove il vincolo di impiego del PC e si inserisce in un ecosistema esistente (Gammataouch).
- L'applicazione Android offre **un'intuitiva interfaccia grafica** per eseguire l'analisi.



Il protocollo GSCP

- Un **messaggio** è rappresentato in memoria da diversi campi:

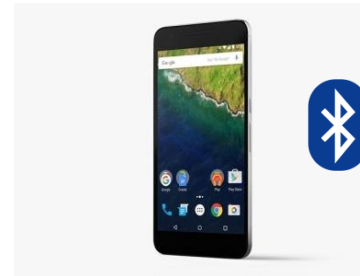
- I messaggi vengono **serializzati** secondo le specifiche del protocollo prima di essere inviati.

```
int tipo = 10;  
int lunghezza = 7;  
String contenuto =  
    pluto 3;
```

Serializzazione

10#7#pluto 3

10#7#pluto 3



10#7#pluto 3

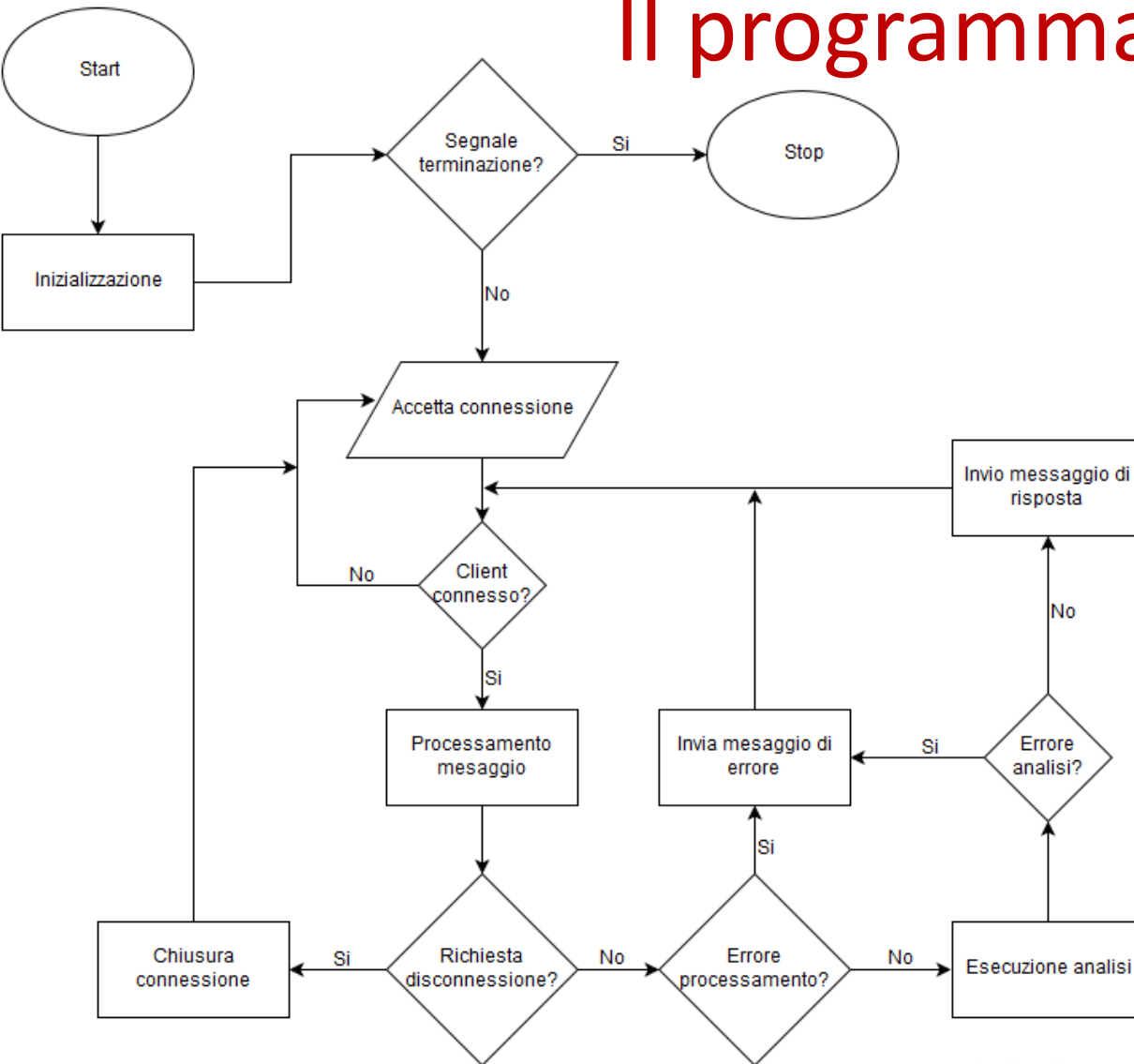
Deserializzazione

```
int type = 10;  
int msgLen = 7;  
char[8] msg =  
    pluto 3;
```

- La **deserializzazione** ricostruisce il **messaggio** a partire dai dati ricevuti.



Il programma GammaServer



Viene eseguito a bordo di yStream in modalità demone.

- **Comunica** con GammaClient attraverso il GSCP.
- **Esegue** le richieste di analisi che riceve dal Client.
- **Risponde** al Client con i risultati delle operazioni effettuate, oppure con un messaggio di errore.

10#7#pluto 3



Ricezione richieste



Processing

11#8#pippo 4



Invio risultati



GammaClient per PC

Disconnect

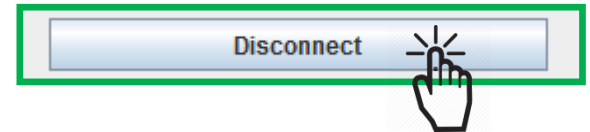
Nome Analisi		Geometria (in metri)		Posizione Detector	
Prefisso File	pappo	Tipo	0	Distanza lungo L1	2
Run ID	3	Lato 1 (L1)	10	Distanza lungo L2	3
		Lato 2 (L2)	10	Altezza	1
		Lato 3 (L3)	0		

Go1

Go2

dT [s]	
m [KeVich]	
Tot 450 - 2900	
Tot 2900 - 6000	
K' [1.37-1.57] cps	
U' [1.66-1.86] cps	
Th' [2.41-2.81] cps	

a	0.373003
b	0.360251
c	0.342126
K [cps]	2.30255
U [cps]	0.424015
Th [cps]	0.332033

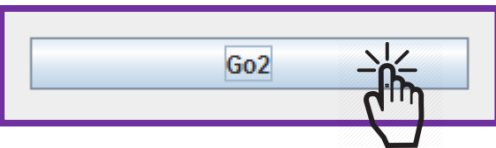


✓ Gestione connessione con γ Stream

✓ Inserimento dei parametri utili all'analisi:

- Nome del file
- Geometria

Nome Analisi		Geometria (in metri)		Posizione Detector	
Prefisso File	pluto	Tipo	0	Distanza lungo L1	2
Run ID	3	Lato 1 (L1)	10	Distanza lungo L2	3
		Lato 2 (L2)	10	Altezza	1
		Lato 3 (L3)	0		

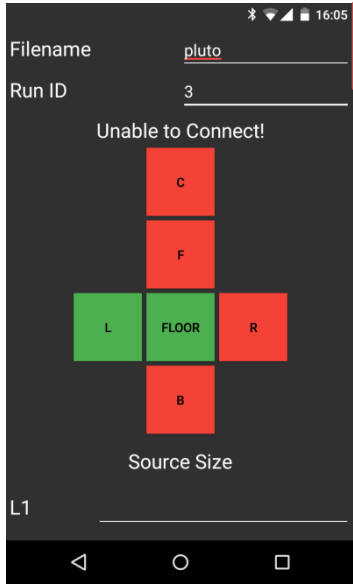


✓ Esecuzione analisi

a	0.373003
b	0.360251
c	0.342126
K [cps]	2.30255
U [cps]	0.424015
Th [cps]	0.332033

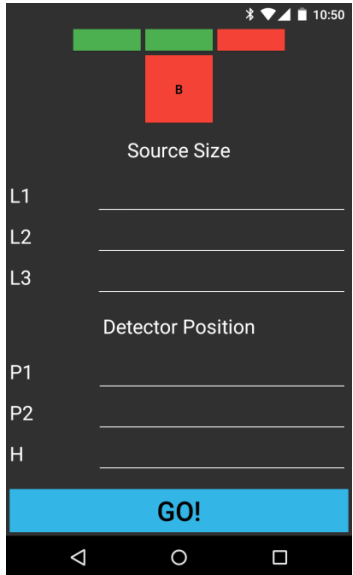
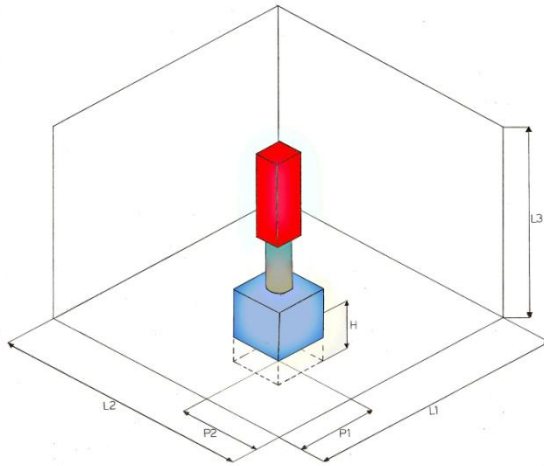
✓ Visualizzazione dei risultati

GammaClient per Android



Interfaccia di Input

✓ Interfaccia utente intuitiva



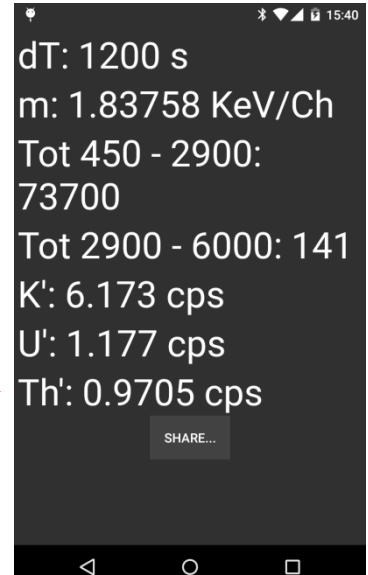
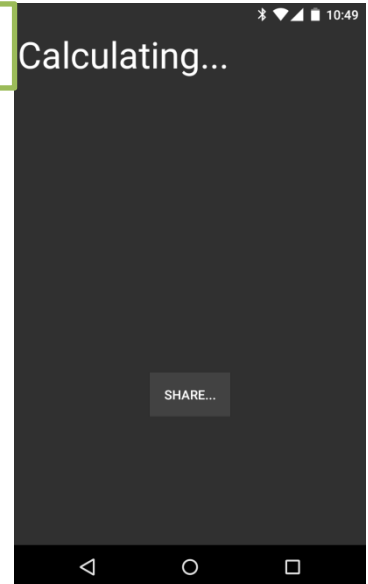
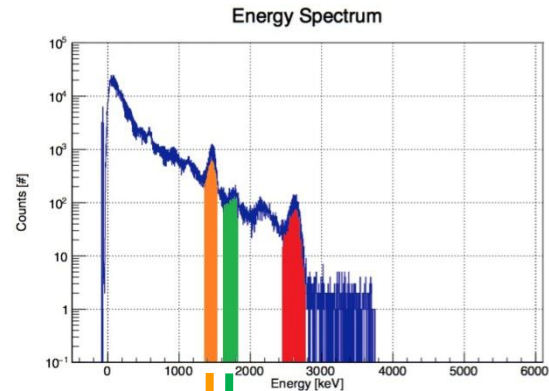
10#7#pluto 3



11#42#1200 1.83758 73700 141
6.173 1.177 0.9705

Interfaccia di Output

✓ Visualizzazione dei counts per second (cps)

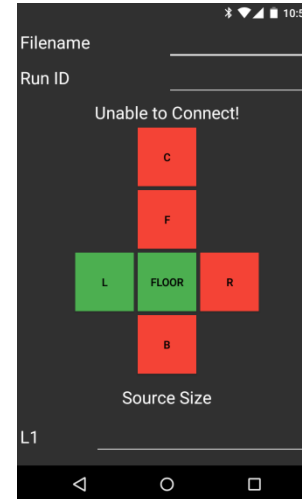


Multi-Threading

Le App Android sono composte di default da un solo thread denominato **UI Thread**.
Lo **UI Thread** gestisce l'interfaccia grafica.

```
UI Thread {  
  button.setColor("green") }
```

Tempo di attesa

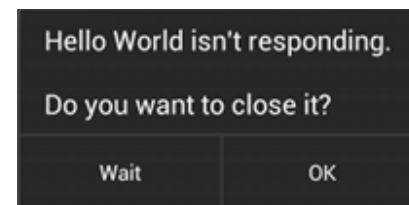
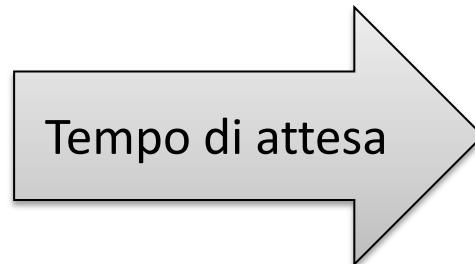


Funzioni dai brevi
tempi di
esecuzione
mantengono fluida
l'interazione con
l'utente...

Lo **UI Thread** non dovrebbe mai eseguire funzioni che richiedono lunghi tempi di elaborazione oppure funzioni bloccanti.

```
UI Thread() {  
  connessione.connect()  
  connessione.read() }
```

Tempo di attesa



...le funzioni
bloccanti no!

Multi-Threading

L'esecuzione dell'App GammaClient è caratterizzata da 3 thread:

- UI Thread
- Producer Thread
- Consumer Thread

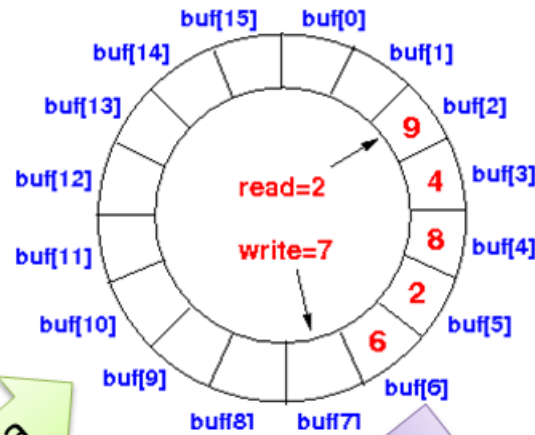
I thread sono divisi in due categorie:

- Produttori
- Consumatori

• I thread comunicano attraverso un **buffer circolare**.

UI Thread
Gestisce la UI

Deposita



In questo modo lo UI Thread non esegue mai **funzioni bloccanti**.

Deposita

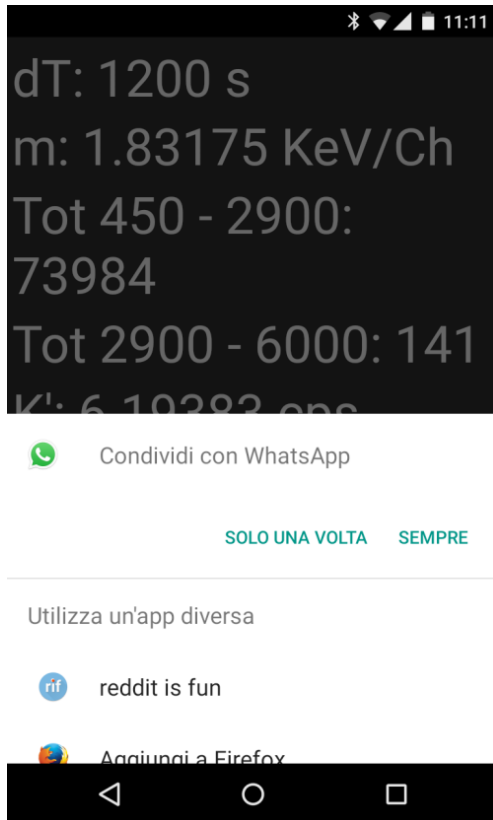
Preleva

Producer Thread
Riceve i messaggi dal server:
`connessione.read()`

Consumer Thread
Esegue le richieste depositate dagli altri thread
Invia i messaggi al Server: `connessione.write()`
Gestisce la connessione: `connessione.connect()`

Condivisione dei risultati

- ✓ Salvataggio sul filesystem
- ✓ Condivisione risultati



La possibilità di condividere i risultati dell'analisi al momento dell'acquisizione dati con un operatore remoto contribuisce ad aumentare l'interazione e la velocità con cui le misure si integrano con un ecosistema più esteso.

Conclusioni e prospettive

Tutti gli obiettivi di questa tesi sono stati raggiunti.

- ✓ Non è più necessario spostare i dati: l'analisi viene eseguita a bordo di γ Stream.
- ✓ Il protocollo Bluetooth implementato elimina l'impiego del cavo RJ-45.
- ✓ L'analisi viene configurata ed eseguita attraverso un'interfaccia grafica.

Prospettive per il futuro.

- Ottimizzare l'interfaccia grafica per altre configurazioni
- Integrare GammaClient con GammaTouch
- Implementare il Service Discovery Protocol (SDP) per la gestione del Bluetooth