

SISTEMI INTEGRATI PER IL MONITORAGGIO, L'EARLY WARNING
E LA MITIGAZIONE DEL RISCHIO IDROGEOLOGICO
LUNGO LE GRANDI VIE DI COMUNICAZIONE

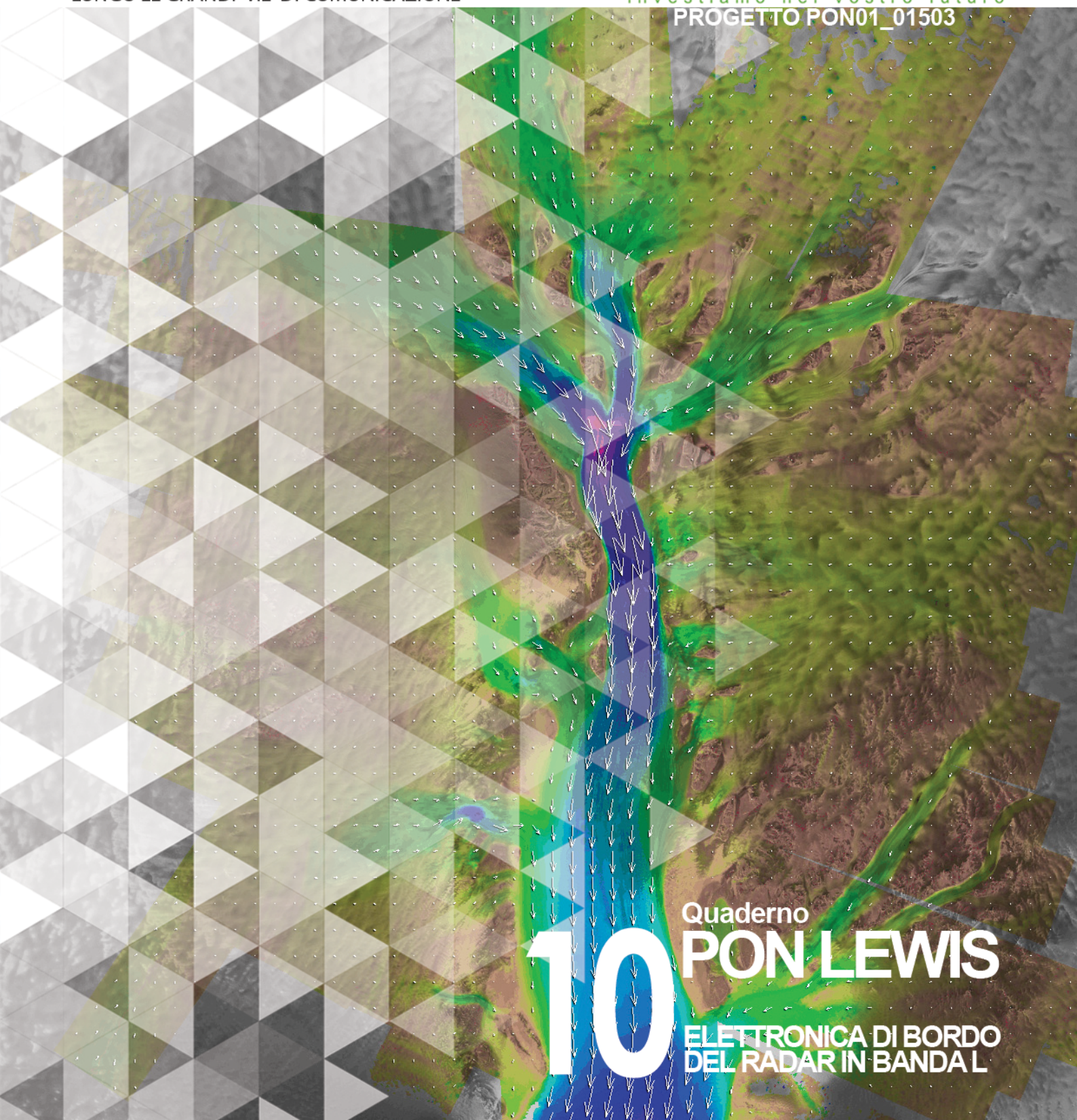


UNIONE EUROPEA
Fondo europeo di sviluppo regionale



investiamo nel vostro futuro

PROGETTO PON01_01503



Quaderno
10 PON LEWIS
ELETTRONICA DI BORDO
DEL RADAR IN BANDA L



autostrade//Tech



A cura di Pasquale Corsonello | **DELIVERABLE WP 3.5**
Elettronica di bordo del radar in banda L

Sistemi integrati per il monitoraggio, l'early warning e la mitigazione del rischio idrogeologico lungo le grandi vie di comunicazione"

Premessa

Frane e inondazioni sono un problema di grande rilevanza nel nostro Paese. Negli ultimi anni le vittime e i danni dei disastri idrogeologici hanno raggiunto livelli inaccettabili e impongono un grande e immediato impegno della comunità nazionale per cercare di mitigare il livello di rischio, utilizzando strategie articolate ed efficaci capaci di integrare, in una visione organica, interventi strutturali e non strutturali.

Su questi temi l'Università della Calabria è impegnata da anni in attività di studio e di ricerca di rilevanza nazionale e internazionale e nella diffusione e promozione della cultura della previsione e prevenzione del rischio idrogeologico. Nel 2011 insieme ad altri partner, ha promosso un progetto di ricerca triennale, "Sistemi integrati per il monitoraggio, l'early warning e la mitigazione del rischio idrogeologico lungo le grandi vie di comunicazione", finalizzato allo sviluppo di un sistema complesso e articolato di preannuncio delle frane da impiegare per le fasi di previsione/prevenzione del rischio idrogeologico.

Il Progetto, indicato con l'acronimo LEWIS (Landslide Early Warning Integrated System), è stato svolto, nel periodo 2012-2014, nel quadro del Programma Operativo Nazionale 2007-13 "Ricerca e Competitività".

I risultati conseguiti sono descritti in questa collana di Quaderni PON LEWIS.

Il progetto è stato sviluppato dall'Università della Calabria e Autostrade Tech S.p.A. insieme ai partner industriali Strago e TDGroup, alle Università di Catania, di Reggio Calabria e di Firenze e al CINID (Consorzio Interuniversitario per l'Idrologia). Per l'Ateneo calabrese hanno partecipato diversi laboratori e gruppi di ricerca: CAMILab (con funzione di coordinamento), μ Wave, Geomatica, Nems, Geotecnica, Dipartimento di matematica.

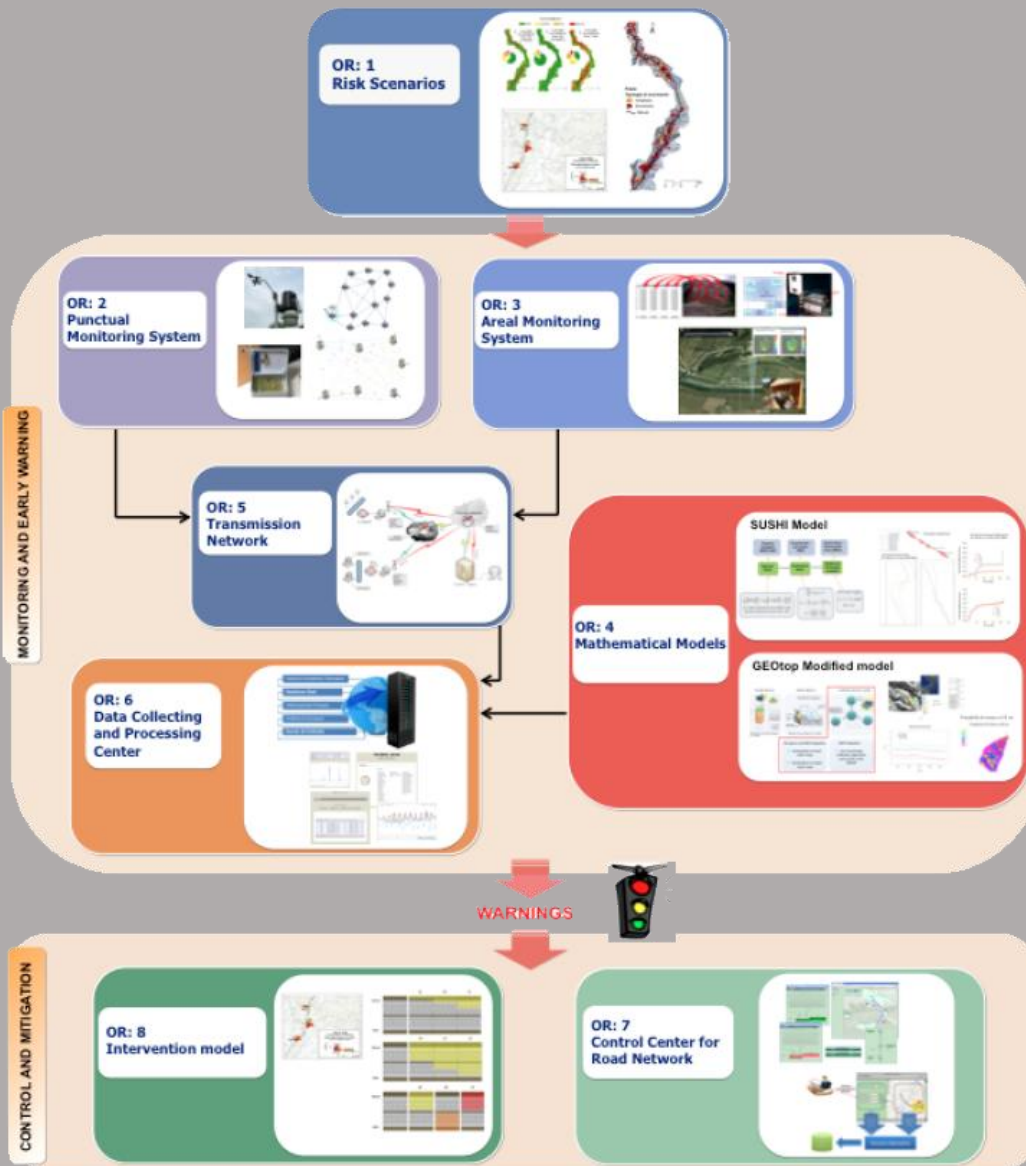


Figura 1 - Articolazione del sistema integrato di monitoraggio dei versanti e di preannuncio dei movimenti franosi

Il progetto è finalizzato allo sviluppo di un sistema di monitoraggio dei versanti e di preannuncio dei movimenti franosi che possono interessare le grandi vie di comunicazione e all'identificazione dei conseguenti interventi non strutturali di mitigazione.

Il sistema è articolato in due sottosistemi (fig. 1):

- ✓ Monitoraggio e preannuncio,
- ✓ Controllo e mitigazione,

che richiedono la preventiva individuazione degli scenari di rischio ossia dei danni che l'eventuale attivazione di una frana può produrre sugli elementi a rischio presenti (infrastruttura viaria, autoveicoli, persone). La procedura originale sviluppata nell'ambito del progetto prevede l'identificazione, lungo il tratto autostradale di interesse, delle aree soggette a movimenti franosi e la conseguente definizione dei relativi scenari di evento e di rischio.

Il sottosistema *Monitoraggio e preannuncio* è formato da diverse componenti: rete di monitoraggio "puntuale" che comprende sensori che misurano localmente l'inizio degli spostamenti superficiali o profondi; rete di monitoraggio "areale" che include sensori che controllano a distanza il fenomeno franoso con tecniche radar; modelli matematici di simulazione dell'innesco e della propagazione dei movimenti franosi. Nel progetto LEWIS sono state sviluppate numerose componenti innovative e sono state modificate e migliorate altre componenti già esistenti. In particolare tra i sensori puntuali sono stati sviluppati i sistemi SMAMID e POIS; tra quelli areali sono stati realizzati un radar in banda L, uno scatterometro, un interferometro; tra i modelli si sono sviluppati e/o migliorati: GEOtop, SUSHI, SCIDDICA.

La raccolta dei dati misurati dai sensori è affidata ad un unico sistema di trasmissione dati che trasmette anche le informazioni necessarie per il funzionamento dei modelli. Il sottosistema è completato da un Centro di acquisizione ed elaborazione dei dati (CAED) che, sulla base dei dati misurati dai sensori e delle indicazioni dei modelli, valuta la situazione di pericolo lungo il tronco autostradale emettendo i relativi livelli di criticità.

I livelli di criticità emessi dal CAED sono l'elemento di collegamento tra il sottosistema *Monitoraggio e preannuncio* e il sottosistema *Controllo e mitigazione*. Gli avvisi di criticità sono acquisiti dal Centro di comando e controllo del traffico (CCCT) che, sulla base di un modello di intervento predefinito, attiva le procedure standardizzate per la mitigazione del rischio, che vanno dalla

sorveglianza diretta del tratto di interesse da parte di squadre tecniche all'interruzione del traffico su entrambe le direzioni di marcia.

Il progetto prevede anche lo sviluppo di attività sperimentali su tre tronchi autostradali lungo la A3, la A16 e la A18, nonché l'erogazione di un Master di secondo livello denominato ESPRI (ESperto in Previsione/Prevenzione Rischio Idrogeologico).

Il progetto di ricerca è stato organizzato in Obiettivi Realizzativi (OR), ciascuno dei quali suddiviso in Work Package (WP), a loro volta articolati in Attività Elementari (AE). In totale erano previste 11 OR, 47 WP e 243 AE. In particolare le OR 1-8 riguardano la ricerca e si articolano in 26 WP e 139 AE. Le OR 9-11 sono dedicate a sperimentazione, governance e trasferimento tecnologico, integrazione e aggiornamento dell'attività di ricerca nella fase di Sviluppo Sperimentale e si articolano complessivamente in 21 WP e 104 AE.

I Quaderni che compongono questa collana sono stati costruiti con riferimento ai singoli WP, per la parte che riguarda la ricerca, e quindi ogni Quaderno contiene la descrizione dei risultati conseguiti nel WP, articolata in base alle AE previste.

Sono, inoltre, previsti altri tre Quaderni:

Quaderno 0 che contiene una descrizione di sintesi, in inglese, dei risultati conseguiti nell'ambito del progetto.

Quaderno 28 che contiene l'informazione relativa alle attività di divulgazione dei risultati scientifici.

Quaderno 29 che contiene la descrizione dei risultati conseguiti con l'attività formativa.

Il Quadro editoriale complessivo è riportato in tabella 1:

QUADERNO	OR	WP	TITOLO
0	-	-	Research outcomes
01 Parte prima	1	1.1	Linee guida per l'identificazione di scenari di rischio
01 Parte seconda	1	1.1	Linee guida per l'identificazione di scenari di rischio
02	2	2.1	Monitoraggio idrogeologico
03 Parte prima	2	2.2	Monitoraggio con unità accelerometriche (Sistema SMAMID)
03 Parte seconda	2	2.2	Monitoraggio con unità accelerometriche (Sistema SMAMID)
04	2	2.3	Circuiti integrati a bassa potenza per sistemi di monitoraggio con unità accelerometriche
05	2	2.4	Monitoraggio con sensori puntuali di posizione e inclinazione (Sistema POIS)
06	3	3.1	Sviluppo di uno scatterometro a risoluzione variabile
07	3	3.2	Elettronica di bordo dello scatterometro ed inclinazione
08	3	3.3	Sviluppo di un radar in banda L
09	3	3.4	Tecniche di analisi e sintesi di segnali radar per la simulazione accurata di scenari complessi
10	3	3.5	Elettronica di bordo del radar in banda L

QUADERNO	OR	WP	TITOLO
11	3	3.6	Sistemi interferometrici radar ad apertura sintetica basati a terra
12	4	4.1	Modello areale per il preannuncio delle frane da innesco pluviale (Modello GEOtop)
13	4	4.2	Modelli completi di versante di tipo puntuale per il preannuncio di movimenti franosi (Modello SUSHI)
14	4	4.3	Modelli di propagazione delle frane tipo colate (Modello SCIDDICA)
15	5	5.1	Rete Wireless di Telecomunicazioni: sviluppo e scelta dei parametri di progetto
16	6	6.1	CAED. Acquisizione dati: architettura del sistema
17	6	6.2	CAED. Elaborazione dei dati
18	7	7.1	CCCT. Progettazione
19	7	7.2	CCCT. Interfaccia verso il centro di acquisizione ed elaborazione dati
20	7	7.3	CCCT. Interfaccia con altre centrali operative e canali di diffusione delle notizie
21	7	7.4	CCCT. Modulo per la presentazione e convalida delle allerte
22	7	7.5	CCCT. Modulo per la gestione delle informazioni di traffico
23	7	7.6	CCCT. Integrazioni con moduli speciali

QUADERNO	OR	WP	TITOLO
24	8	8.1	Definizione del modello di intervento e predisposizione del Piano di Emergenza
25	8	8.2	CCCT. Gestione delivery allerte e attivazione squadre d'intervento
26	8	8.3	CCCT. Gestione percorsi alternativi
27	9	9.1 - 9.11	Sperimentazione
28	10	10.1 - 10.2	Piano di comunicazione e diffusione
29	-	-	Master ESPRI (Esperto in Previsione/Prevenzione Rischio Idrogeologico)

Tabella 1 - Quadro editoriale complessivo della collana di Quaderni PON LEWIS

31 dicembre 2014

Il Responsabile Scientifico del progetto PON LEWIS

Pasquale Versace

INDICE

ATTIVITA' ELEMENTARE 3.5.1

1 **Deliverable Fisico**

ATTIVITA' ELEMENTARE 3.5.2

2 **Deliverable Documentale**

2 **3.5.2.1 Sistema di controllo**

3 **3.5.2.2 Hardware utilizzato**

6 **3.5.2.3 Connessione alla rete telefonica ed internet**

7 **3.5.2.4 Esempio di elaborazione**

9 **3.5.2.5 Elenco comandi e configurazioni e-mail/sms**

10 **3.5.2.6 Sviluppo software SBC**

28 **3.5.2.7 Configurazioni Software SBC**

30 **3.5.2.8 Software embedded PC**

32 **3.5.2.9 Test effettuati**

Introduzione

L'attività relativa al WP3.5 ha riguardato lo studio di soluzioni adatte alla realizzazione di un sistema di controllo remoto per un radar in banda P. Il sistema progettato e realizzato consente l'attivazione delle procedure di accensione e spegnimento degli apparati attraverso comandi ricevuti via SMS ed email, e la trasmissione dei dati di output del radar.

ATTIVITÀ ELEMENTARE 3.5.1

Deliverable Fisico

L'attività si è conclusa con la realizzazione del prototipo illustrato in Figura 0. Il cuore del sistema è costituito dal SBC Rapsberry. Il prototipo è stato consegnato ai Colleghi del Gruppo di Elettromagnetismo del DIMES-UNICAL che hanno effettuato le successive operazioni di integrazione in collaborazione con i membri del Gruppo di Ricerca.



Figura 0

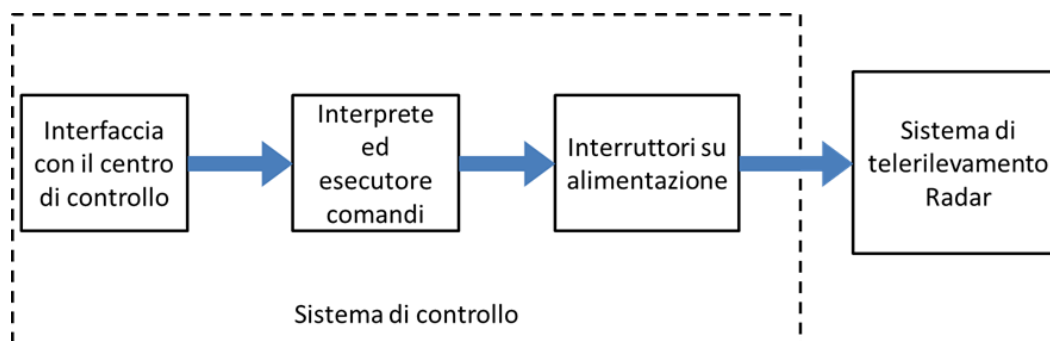
Deliverable Documentale

Il presente Deliverable ha lo scopo di descrivere le varie sezioni del sistema di controllo per il radar in banda P. Poiché nel sito di installazione non è prevista la presenza costante di un operatore, è apparso necessario sviluppare una struttura di supporto al funzionamento dell'applicazione dedicata all'accensione e allo spegnimento controllato degli elementi componenti il sistema. Tale sistema risulta composto da diversi elementi, quali antenne, amplificatori di potenza a RF, embedded PC ecc., che necessitano di notevoli potenze di alimentazione e che sono in grado di dissipare elevato calore. Pertanto il sistema di telerilevamento non potrà essere attivo di continuo, ma dovrà essere messo in funzione soltanto il tempo necessario per compiere un rilevamento e inviare i dati raccolti ad un centro di elaborazione dati.

3.5.2.1 SISTEMA DI CONTROLLO

A livello logico, è possibile schematizzare il sistema di controllo suddividendolo in tre blocchi funzionali:

- Interfaccia con il centro di controllo
- Interprete ed esecutore comandi
- Interruttori



L'**interfaccia con il centro di controllo** si occupa di ricevere i comandi da remoto e successivamente di inoltrare questi al blocco dedicato all'interpretazione.

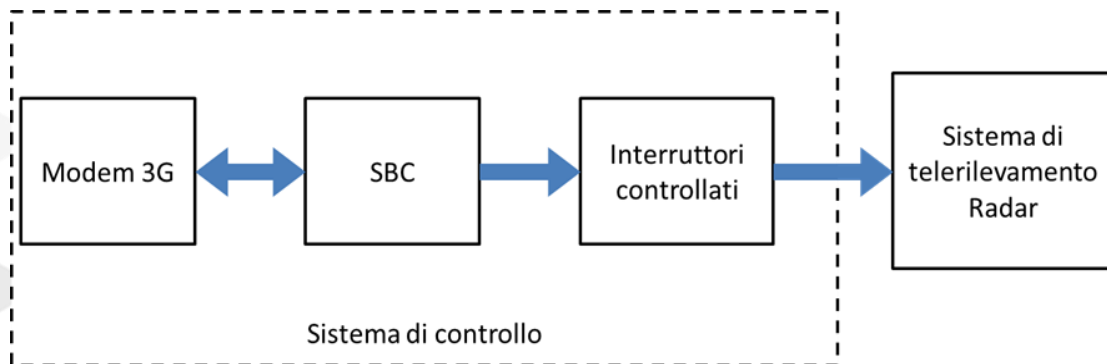
L'**interprete ed esecutore dei comandi** si occupa di valutare la correttezza e la validità del comando. Una volta accertato ciò, interpreterà il comando facendo riferimento ad una tabella di comandi ammissibili e successivamente agirà sugli interruttori.

Gli **interruttori**, interposti sulla linea di alimentazione, qualora attivati dal blocco precedente, forniranno energia elettrica al sistema di telerilevamento, in modo tale che questo possa essere attivato.

Attraverso la rete telefonica mobile il centro di controllo può inviare comandi al sistema di controllo remoto attraverso SMS ed e-Mail. L'interfaccia con il centro di controllo prevede l'utilizzo di un modem 3G. E' stato quindi necessario utilizzare una **single-board-computer** (SBC) con interfacce di connessione standard e dotato di un sistema operativo in grado di poter gestire tale modem.

3.5.2.2 HARDWARE UTILIZZATO

In base a quanto sopra discusso, si è pervenuti alla seguente schematizzazione del sistema.



Di seguito vengono riportati i componenti hardware scelti.

- **HSUPA Modem:** è un modem HSDPA provvisto di interfaccia USB. Il dispositivo è utilizzato per connettersi a Internet utilizzando le reti di telefonia mobile 3.5G , 3G o 2G. Supporta UMTS (tra cui HSDPA), EDGE, GPRS e GSM.



- **SBC Raspberry:** è un SBC (Single Board Computer) dotata di un processore ARM e 256 MB di memoria RAM, interfacce Ethernet,USB, user Digital I/O, DVI. È dotata di sistema operativo Linux. Grande come una carta di credito, ma molto funzionale, è perfetta per portare capacità elaborativa in posti inusuali dove lo spazio e l'elettricità sono una risorsa preziosa. Fra gli altri vantaggi c'è il prezzo particolarmente modesto.

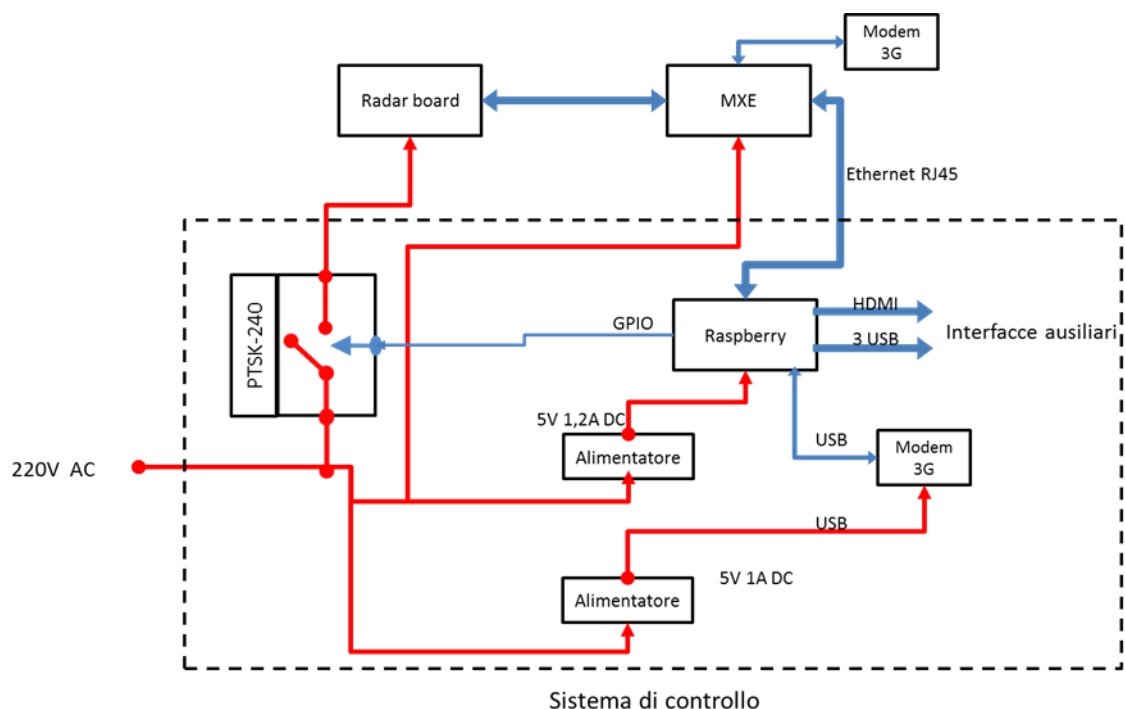


- **PSTK-240:** è un kit che permette la configurazione di un “Power Switch” adatto per il controllo di corrente alternata con microcontrollori o altri circuiti a bassa tensione. Nel caso specifico è stato configurato per interrompere il collegamento con la rete elettrica standard in uso in Italia . Questo kit è basato sul relè Zettler AZ21501. E’ possibile controllare tale relè tramite un segnale in bassa tensione a 3.3V, otticamente isolato dalla parte del circuito connessa alla rete. Il relè supporta circa 250.000 cicli con correnti di 20 A. Di seguito è illustrata un’immagine del PSTK-240.



Diagramma a Blocchi del Sistema di Controllo

Come mostrato nella figura seguente, si è scelto di utilizzare approcci differenti per il controllo dei diversi elementi del sistema di telerilevamento. In particolare la **radar board** verrà alimentata/disalimentata esattamente come ipotizzato in precedenza, mentre l’embedded PC che governa questa (**MXE**) è controllato via software dalla SBC, attraverso una rete lan cablata secondo lo standard Ethernet.



E' infine opportuno precisare che lo MXE comunicherà con il centro di elaborazione dati attraverso un connessione 3G differente rispetto a quella utilizzata dal sistema di controllo.

3.5.2.3 CONNESSIONE ALLA RETE TELEFONICA ED INTERNET

Il sistema di controllo, come evidenziato nella parte introduttiva, può ricevere comandi dall'utente remoto attraverso la rete telefonica, mediante la ricezione di SMS, o tramite la rete internet, mediante la ricezione di e-mail. L'interfaccia per il sistema di controllo verso queste reti è il HSUPA 3G. Questo modem, attraverso la connessione alla rete internet, permette la ricezione di e-mail contenente comandi da server di posta POP3, e inoltre consente la ricezione di SMS al numero di telefono corrispondente alla SIM montata al suo interno.

La gestione di questo dispositivo da parte del SBC Raspberry non presenta particolari difficoltà con Linux in quanto è stato inserito il supporto per esso nel kernel Linux a partire dalla versione 2.6.20.

3.5.2.4 ESEMPIO DI ELABORAZIONE

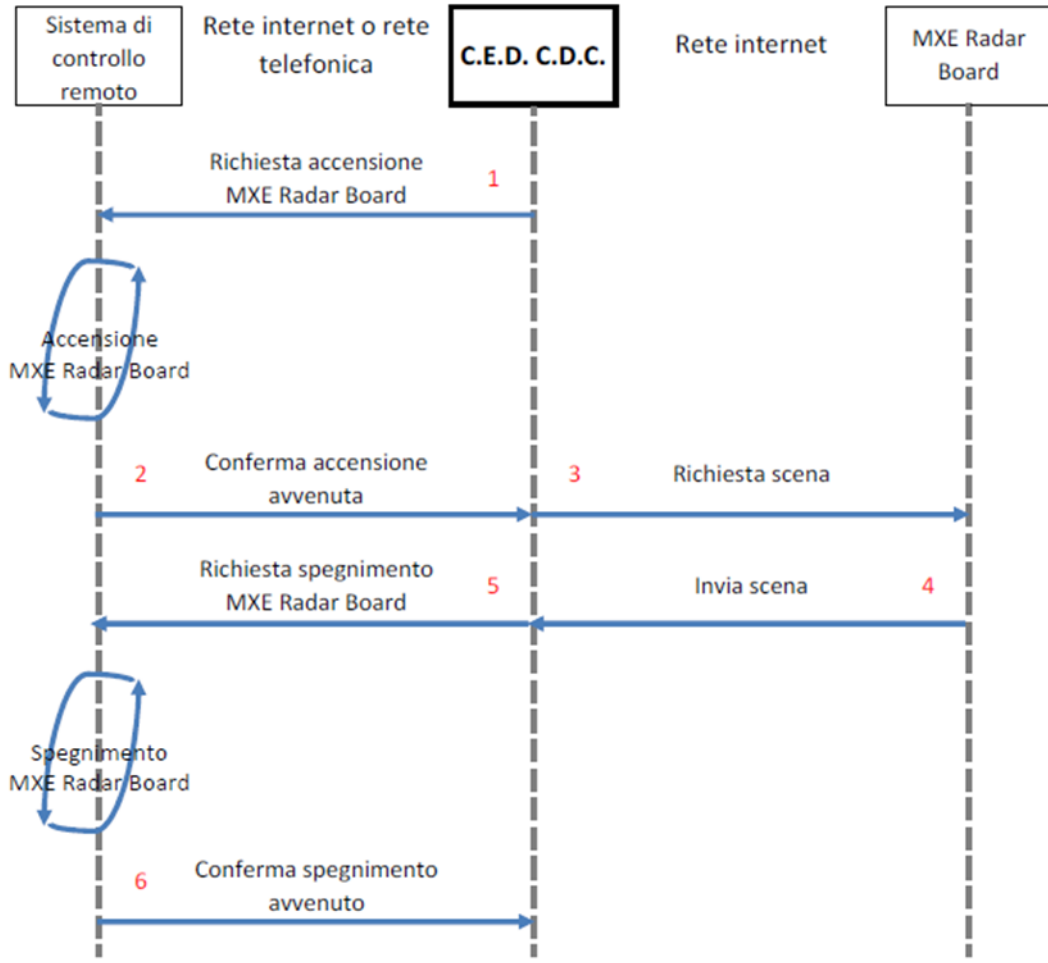
Una volta stabilita l'architettura, scelti i componenti e assemblato il tutto, la definizione dell'hardware può considerarsi conclusa.

Tutte le funzionalità del sistema sono state realizzate attraverso software applicativo da eseguire sulla SBC.

È stato necessario a questo punto fissare quali operazioni devono essere svolte dal sistema e come questo interagisce con il centro di controllo e lo MXE che controlla la radar board.

Di seguito viene riportato come esempio il flusso di elaborazione completo relativo all'acquisizione di una scena. Questo tipo di operazione richiede un primo contatto via SMS/e-mail da parte del centro di elaborazione dati/centro di controllo (**C.E.D. C.D.C.**) al sistema di controllo remoto. Questo provvede ad eseguire il comando di accensione del MXE e alla radar board confermare l'avvenuta esecuzione di tale comando. A questo punto il centro contatta direttamente lo MXE che dispone di un collegamento alla rete internet indipendente e vengono acquisiti e inviati i dati relativi ad una scena. Adesso il centro può ricontattare il sistema di controllo remoto per spegnere MXE e radar board.

Nell'immagine successiva è possibile osservare una schematizzazione di quanto detto.



3.5.2.5 ELENCO COMANDI E CONFIGURAZIONI E-MAIL/SMS

La SBC è connessa attraverso un modem 3G sia alla rete telefonica che alla rete internet; questo le permette di ricevere comandi attraverso l'interpretazione di e-mail/SMS. Ricevuti i testi di tutti le e-mail/ SMS, l'applicazione in esecuzione sulla Raspberry deve cercare all'interno di essi una particolare stringa che contiene eventuali comandi. In particolare la stringa è così composta:

/pass/password//comm/comando//

La stringa contiene due informazioni, ovvero il campo *password* e il campo *comando*. Il *comando* verrà ritenuto valido se *password* risultasse uguale a quella impostata nell'applicazione.

Se all'interno dei testi non ci sono stringhe nel formato valido, oppure se la password non corrisponde, verrà stampato un messaggio che ci avvisa che non sono stati trovati comandi. Altrimenti il comando verrà inviato allo standard output (stdout).

Una volta letti, tutte le e-mail e tutti i messaggi presenti sulla SIM verranno cancellati.

Di seguito viene fornito un elenco dettagliato dei comandi implementati e viene data una loro breve descrizione.

Comando	Descrizione
PWON	Alimentazione ON
PWREBOOT	Alimentazione OFF – ON
PWOFF	Alimentazione OFF
PCON	Accende MXE5303X
PCREBOOT	Spegne e riaccende MXE5303X
PCOFF	Spegne MXE5303X
SYSON	Alimentazione ON Accende MXE5303X
SYSREBOOT	Alimentazione OFF – ON Spegne e riaccende MXE5303X
SYSOFF	Alimentazione OFF Spegne MXE5303X
RASPREBOOT	Riavvia la Raspberry
RASPOFF	Spegne la Raspberry

Un esempio di comando può essere quello indicato di seguito:

`/pass/temppassword//comm/SYSON//`

Di seguito vengono indicati il numero di telefono, l'indirizzo e-mail ecc. per la ricezione dei comandi da parte di questo kit.

- **Numero di telefono:** **+39331 6439 281**
- **Pin:** **9085**
- **Indirizzo e-mail:** **earlywarningls@yahoo.it**
- **Password e-mail:** **earlywar01**
- **Password comandi:** **earlywar01**
- **Indirizzo e-mail risposte:** **XXX@XXX.XX**

Password e-mail risposte: XXXX

3.5.2.6 SVILUPPO SOFTWARE SBC

.....

Il flusso di sviluppo del software per questi sistemi ricalca essenzialmente quelli delle tradizionali applicazioni per desktop anche se con alcune differenze. Ad esempio se si ha la necessità di utilizzare il supporto del sistema operativo può essere necessario strutturare e configurare le sue funzionalità al fine di renderle adatte alle esigenze del software applicativo e dell'hardware.

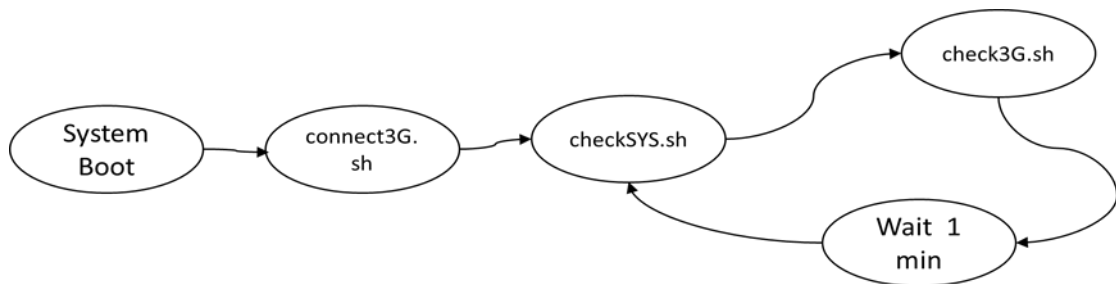
A livello software il sistema sviluppato è composto da diverse applicazioni (sviluppate in C e compilate con *gcc*) e script (per la shell Bourne di Linux) che in esecuzione sulla SBC rendono il tutto in grado di ricevere comandi da e-mail o SMS, accendere e spegnere l'elaboratore controllato, ricevere e inviare dati, configurazioni e comandi ad esso, riavviare la SBC, creare dei LOG sull'attività del sistema e inoltrare eventuali risposte del sistema ad una casella di posta elettronica.

L'esecuzione di queste applicazioni viene lanciato automaticamente all'avvio della SBC e viene monitorato attraverso il servizio *cron* di Unix il quale permette di eseguire una manutenzione automatica del sistema stesso.

L'immagine a seguire mostra il flusso di esecuzione del software lanciato eseguito all'avvio del sistema. Una volta che il sistema ha eseguito il boot, viene avviato lo script `connect3G.sh` che si occupa di stabilire la connessione alla rete internet.

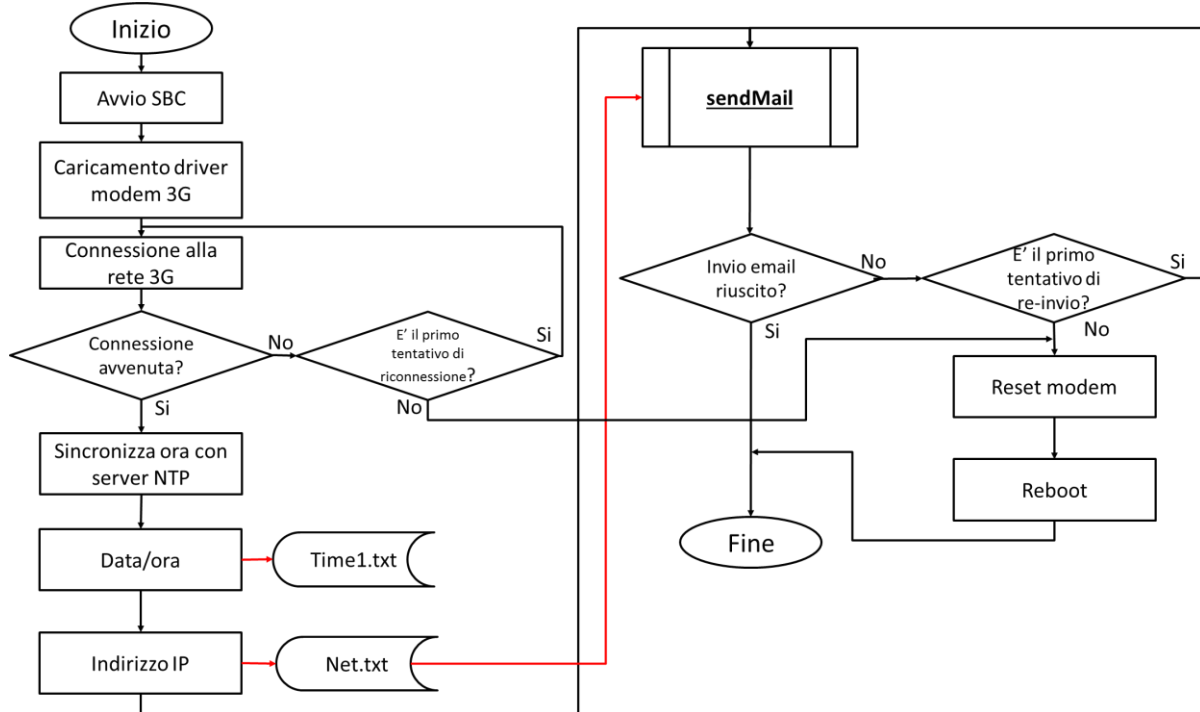
Le parole formattate **parola** indicano routine eseguibili, il cui flusso viene illustrato successivamente.

I sorgenti delle applicazioni, gli eseguibili e i file di configurazione sono tutti contenuti nella directory `/home/pi`.



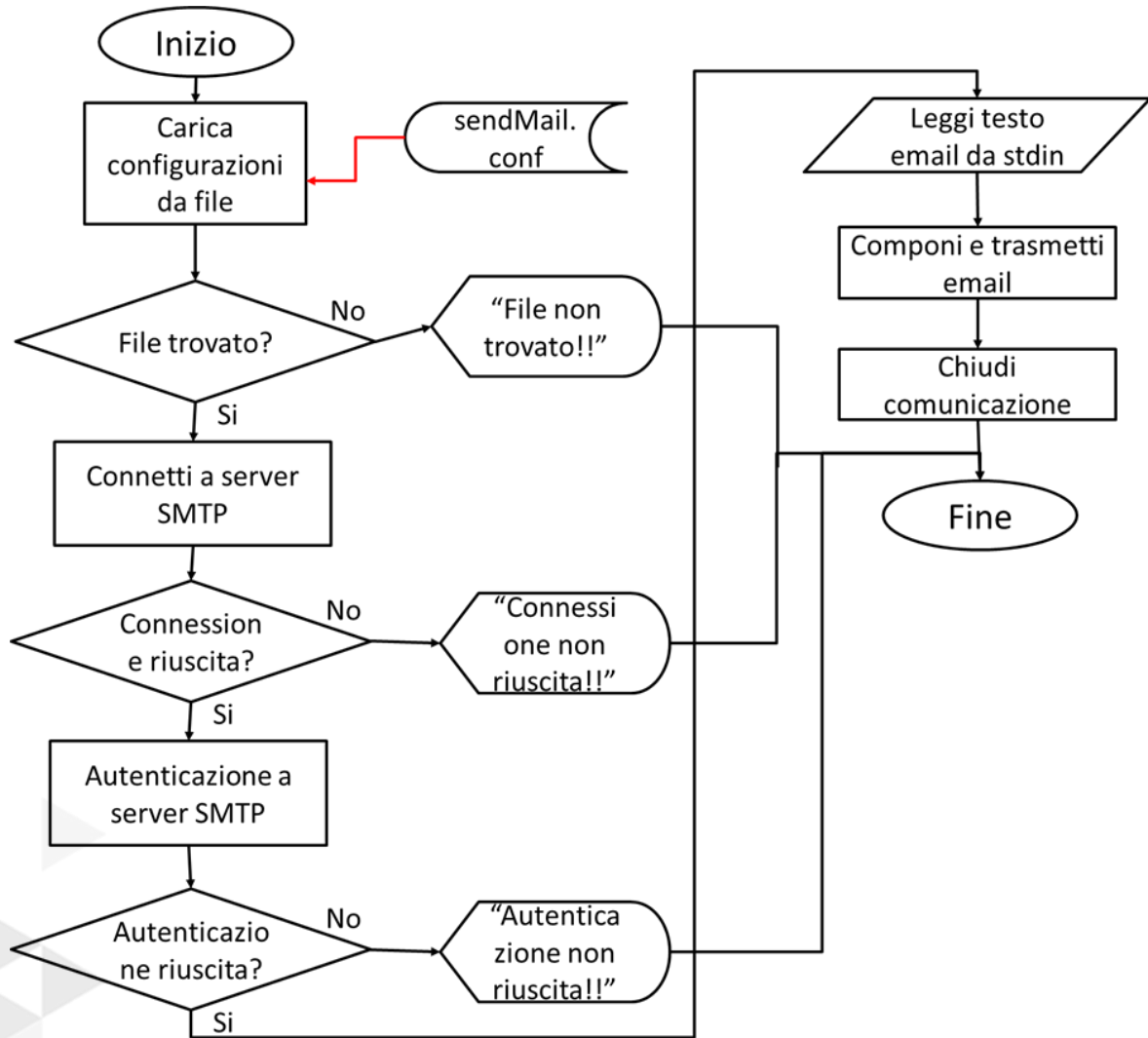
CheckSYS.sh, check3G.sh sono attività ricorrenti con esecuzione pianificata nel sistema operativo. Le distribuzioni Linux hanno infatti un file `/etc/crontab` dove vengono pianificate le attività ricorrenti. Questi script si occupano di lanciare e controllare l'esecuzione di **remoteClient.sh** (l'applicazione principale del sistema, la quale coordina tutte le operazioni che vengono eseguite) e di controllare lo stato della connessione alla rete internet. E' stato pianificato di eseguire questi script ad intervalli di un minuto.

connect3G.sh



All'accensione della board SBC viene lanciato automaticamente lo script **connect3G.sh**. Questo script si occupa per prima cosa di stabilire una connessione alla rete 3G. Se la connessione ha esito positivo, viene sincronizzato l'orologio di sistema mediante il protocollo NTP, e viene salvata l'ora attuale sul file *Time1.txt*. Successivamente viene scritto l'indirizzo IP ottenuto dalla rete 3G nel file *Net.txt* e questo viene inviato come e-mail su una casella di posta predefinita tramite l'applicazione **sendMail**. Qualora il tentativo di connessione o l'invio della e-mail producessero un risultato negativo, l'operazione viene ripetuta una volta. Nel caso anche questo secondo tentativo producesse un risultato negativo, si ipotizza un crash da parte del modem. Viene quindi resettato il modem e riavviato il sistema.

sendMail



L'applicazione **sendMail** è stata sviluppata in C e si occupa di inoltrare ad un server SMTP (Simple Mail Transfer Protocol) una e-mail. Sarà poi questo a inviare l'e-mail all'indirizzo di posta del destinatario. La connessione al server avviene mediante l'utilizzo di socket.

Dopo aver caricato le impostazioni dal file *sendMail.conf*, l'applicazione tenterà di collegarsi al server SMTP che indicato in quest'ultimo.

Di seguito viene illustrato un esempio di questo file.



Username e password utilizzate per l'autenticazione dell'utente sul server SMTP sono codificate (vedi codifica BASE64).

Il software di invio e-mail è di semplice utilizzo ed è accessibile mediante il comando *sendMail*. La sintassi del comando è la seguente **sendMail** *destinatario_e-mail oggetto_e-mail percorso_file_testo_e-mail*.

Stabilita la comunicazione inizia uno scambio di messaggi tra server SMTP e client.

Il primo passo effettuato è l'autenticazione dell'utente sul server. Di seguito viene riportato un esempio.

```
Server)      220 ...
Client) Ehlo hostname
Server)      250 ...
Client) AUTH LOGIN

Server)      334 ...
Client) username_BASE64
Server)      334 ...
Client) password_BASE64
Server)      235 ...
```

Successivamente, dopo aver acquisito dallo standard input (stdin) il testo dell'email, l'oggetto e il destinatario, viene composto e trasmesso l'email.

Client) **MAIL FROM:** <indirizzo_e-mail_sender>
Server) **250** ...
Client) **RCPT TO:** <indirizzo_e-mail_destinatario>
Server) **250** ...
Client) **DATA**
Server) **354** ...
Client) **Subject:** oggetto_e-mail
Client) **From:** indirizzo_e-mail_sender
Client) **To:** indirizzo_e-mail_destinatario
Client)
Client) *Testo e-mail*
Client) **CRNL.CRNL**
Server) **250** ...

Infine viene chiusa la comunicazione.

Client) **QUIT**
Server) **221** ...

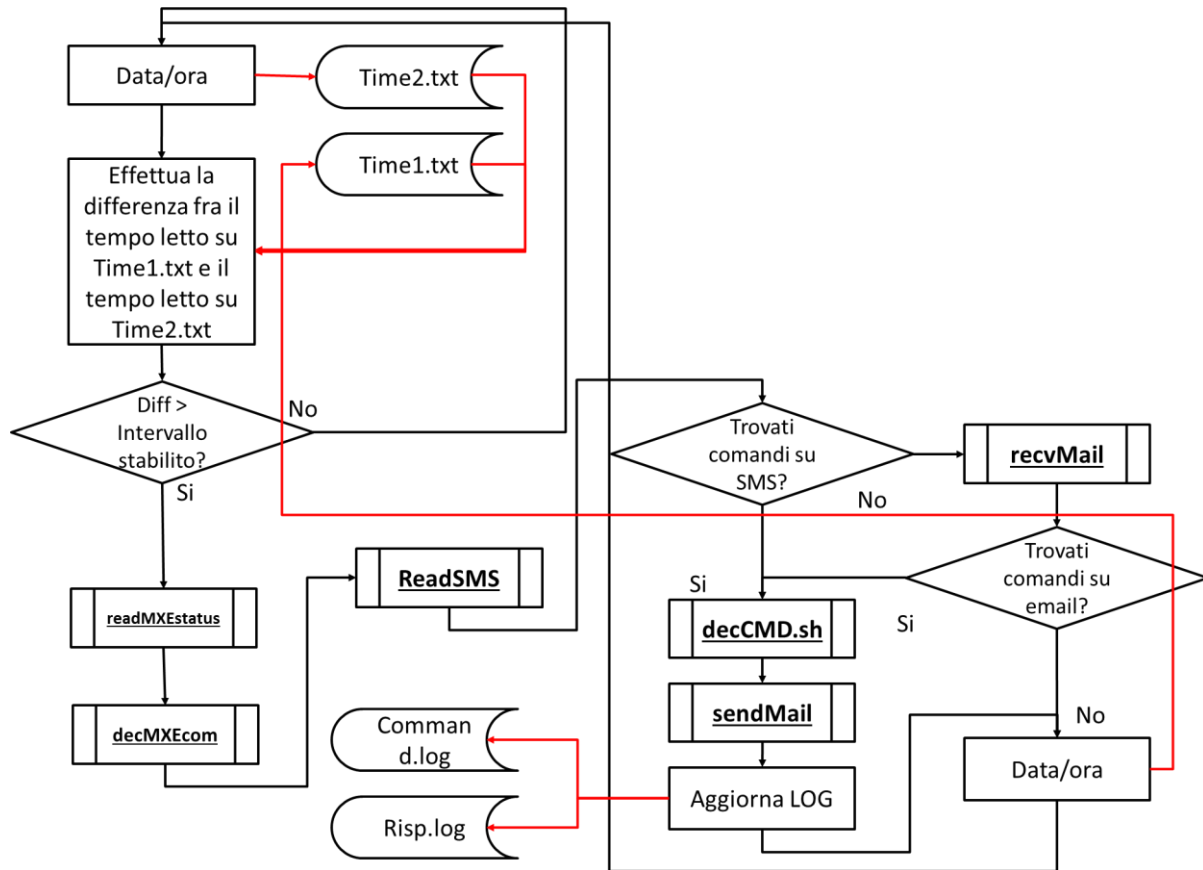
check3G.sh

Questo script controlla lo stato della connessione alla rete internet. Qualora venisse rilevato che la connessione non è attiva, viene rilanciato lo script **connect3G.sh**.

checkSYS.sh

Questo script controlla se **remoteClient.sh** è in esecuzione, qualora non lo fosse lo lancia. Qualora **remoteClient.sh** fosse in esecuzione, ma questa risultasse bloccata, il sistema verrà riavviato.

remoteClient.sh



Lo script **remoteClient.sh** si occupa periodicamente della ricezione di comandi dal server presente sul MXE collegato localmente attraverso Ethernet e di ricevere comandi validi via SMS o email.

L'applicazione **readMXEstatus** si occupa di ricevere comandi dal server presente sul MXE, i quali verranno successivamente interpretati ed eseguiti dallo script **decMXEcom.sh**.

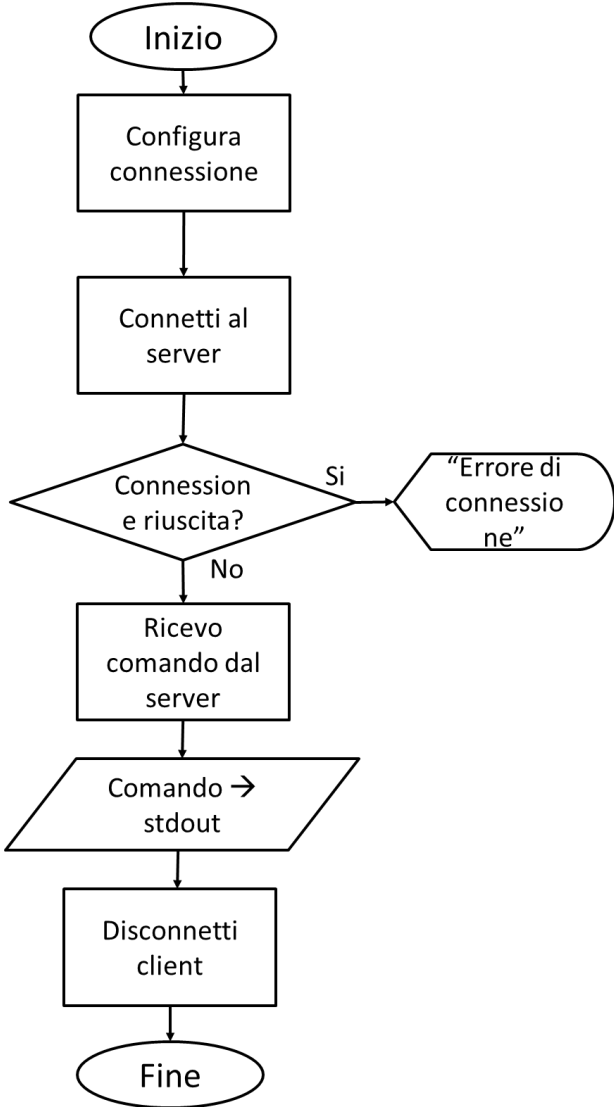
Le applicazioni **ReadSMS** e **recvMail** cercano i comandi validi rispettivamente sulla SIM e sulla casella di posta predefinite.

Se ricevono dei comandi validi da queste applicazioni, questi verranno passati successivamente allo script **decCMD.sh** che si occupa di interpretarli ed eseguirli.

Lo script **decCMD.sh** una volta eseguito il comando, produrrà una risposta ad esso. Questa verrà inoltrata ad una casella email predefinita, che potrà essere consultata al fine di conoscere l'esito del comando inviato. Il comando ricevuto e la sua eventuale risposta verranno salvati inoltre su due file di log (Command.log, Risp.log) posizionati nella cartella */home/pi/log*.

Infine lo script controlla continuamente se è trascorso l'intervallo di tempo stabilito tra un controllo e l'altro, facendo la differenza fra la data/ora attuale e quella salvata al termine del controllo precedente.

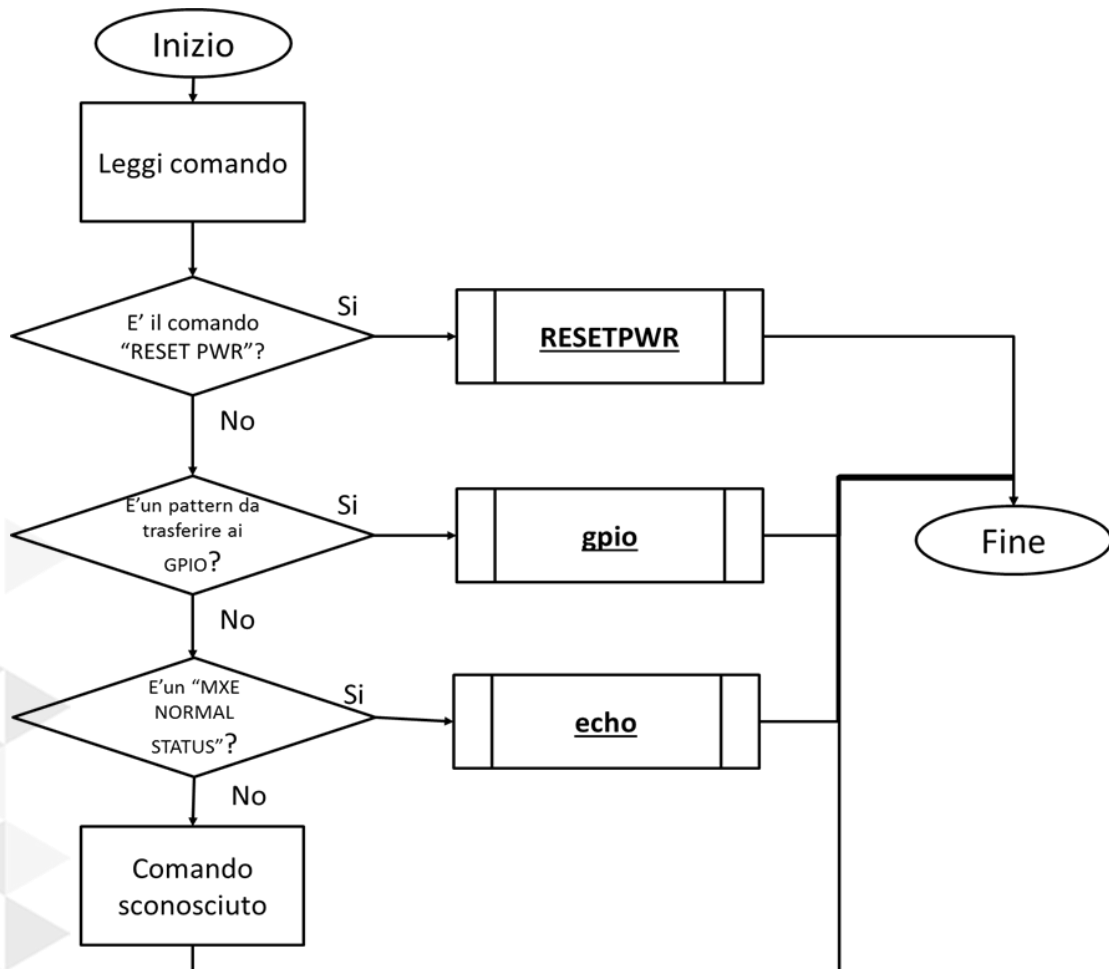
readMXEstatus



E' stato necessario prevedere questa procedura in grado di ripristinare il sistema nell'eventualità di un malfunzionamento o blocco dovuto a cause di varia natura.

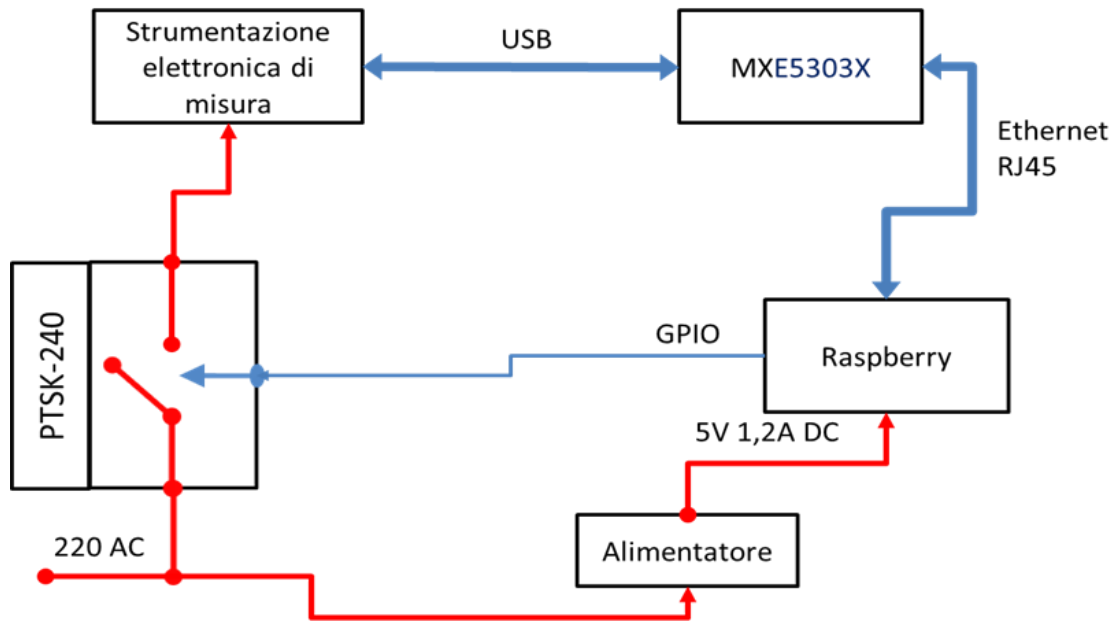
In particolare è stata concentrata l'attenzione su un eventuale blocco della strumentazione elettronica di misura o della comunicazione tra questa e il calcolatore MXE5303X. In questo caso se non si potesse intervenire in nessun modo, il funzionamento dell'intero sistema verrebbe compromesso. E' stato pertanto prevista una comunicazione socket tra l'embedded PC e la SBC Raspberry che indichi a quest'ultima lo stato della strumentazione elettronica. Il servizio sarà attivo sulla porta TCP 4000 dell'embedded PC.

decMXEcom



Lo script **decMXEcom.sh** si occupa di decifrare il comando ricevuto dal server ed eseguirlo. Attualmente sono stati previsti tre comandi:

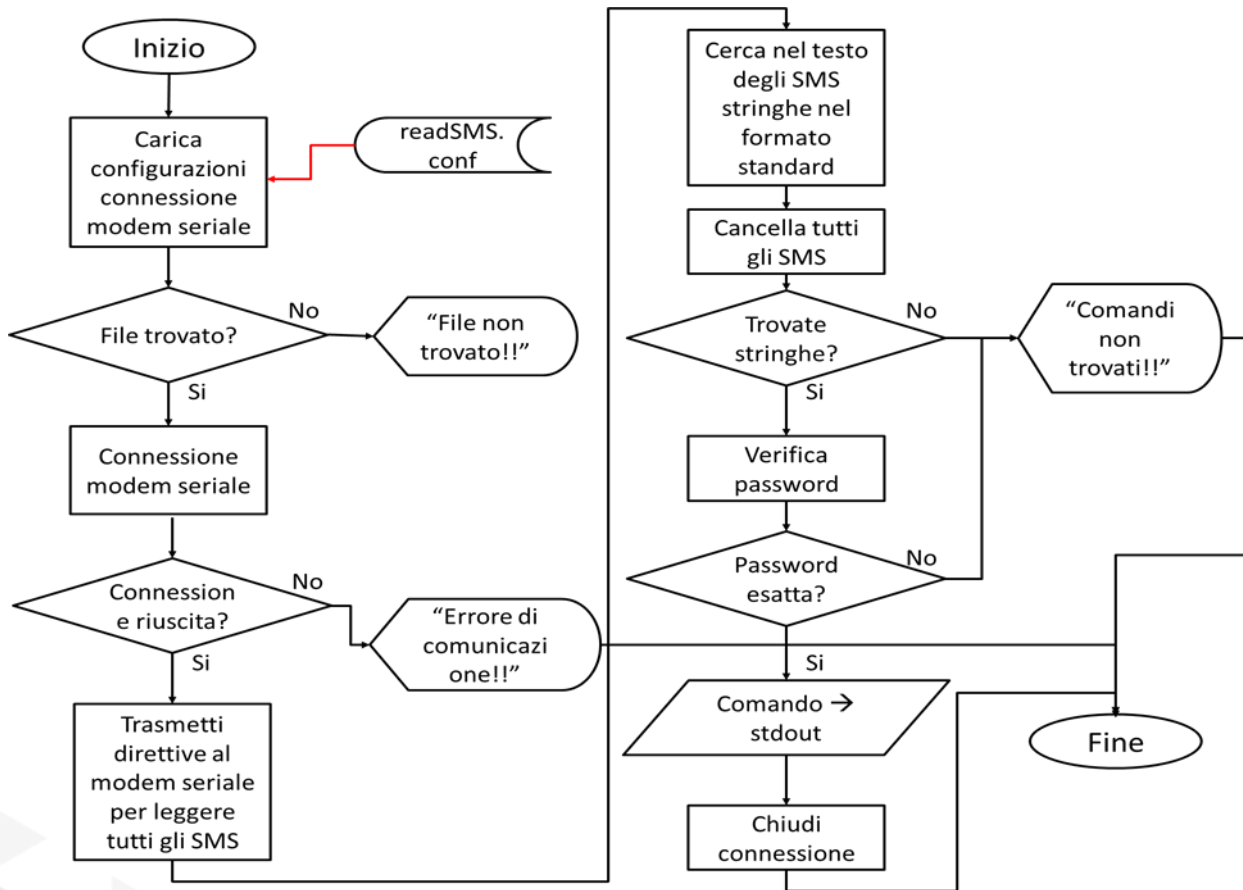
- **RESETPWR**: il server MXE chiede il RESET delle apparecchiature alimentate dal relè. La SBC Raspberry, provvederà a riavviare il sistema interrompendo e successivamente ripristinando l'alimentazione. Ovviamente la strumentazione dovrà prevedere una fase di inizializzazione che sia in grado di riportare i dispositivi allo stato iniziale.



Qualora l'esecuzione della procedura di ripristino non sortisca i risultati sperati, il sistema dovrà richiedere un intervento di manutenzione.

- **MXE NORMAL STATUS**: sull'embedded PC non sono state riscontrate anomalie. La SBC si limita a generare un eco.

readSMS



ReadSMS è un'applicazione sviluppata in C e compilata con il tool *gcc* disponibile per Linux. Questa applicazione si occupa di interfacciarsi con il modem 3G. Con questo modem collegato alla SBC tramite USB, è possibile stabilire un collegamento di tipo seriale. Una volta stabilito il collegamento, l'SBC è in grado di inviare direttive al modem per stabilire una connessione alla rete 3G, ricevere ed inviare SMS, ricevere ed inviare telefonate. Questo standard di direttive prende il nome di "AT Commands". In particolare questa applicazione, come d'altronde il nome suggerisce, utilizza le direttive AT allo scopo di leggere gli SMS inviati alla SIM card contenuta nel modem.

Di seguito viene riportato un esempio che mostra lo scambio di messaggi effettuato per poter leggere tutti gli SMS.

```
SBC)      AT
Modem)    OK
SBC)      AT+CMGF=1
Modem)    OK
SBC)      AT+CPMS="SM"
Modem)    +CPMS ...
SBC)      AT+CMGL="ALL"
Modem)    +CMGL Testo SMS
```

Ricevuti i testi di tutti gli SMS, l'applicazione cerca all'interno di essi una particolare stringa eventuali comandi. In particolare la stringa è così composta:

/init/password/pass/comando//

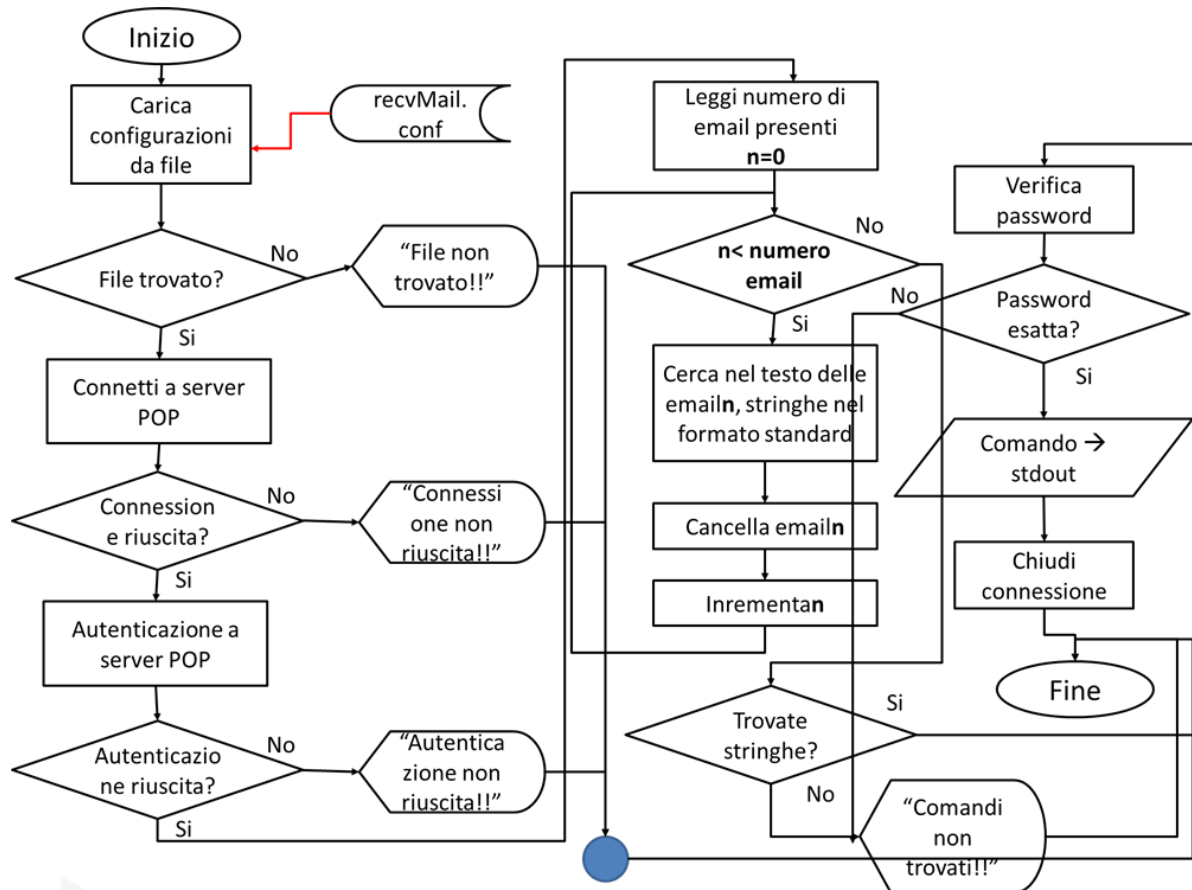
La stringa contiene due informazioni, ovvero il campo *password* e il campo *comando*. Il *comando* verrà ritenuto valido se *password* risultasse uguale a quella impostata nel file *readSMS.conf*.

Se all'interno dei testi degli SMS non ci sono stringhe nel formato valido, oppure se la password non corrisponde, verrà stampato un messaggio che ci avvisa che non sono stati trovati comandi., altrimenti il comando verrà inviato allo standard output (stdout).

Una volta letti, tutti i messaggi presenti sulla SIM vengono cancellati.

```
SBC)      AT+CMGD=1,4
Modem)    OK
```

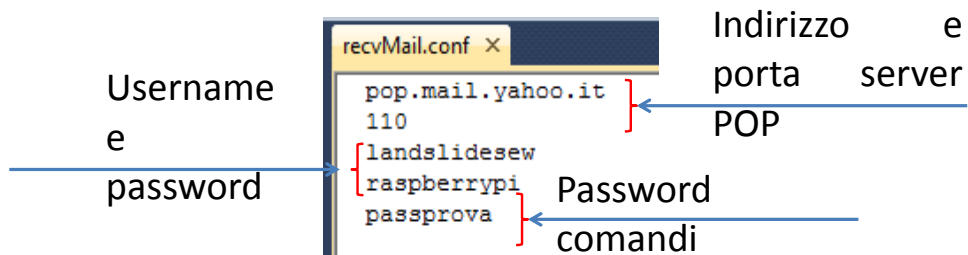
recvMail



L'applicazione **recvMail** è stata sviluppata in C e si occupa di recuperare dal server POP (Post Protocol) le email destinate ad uno specifico indirizzo di posta elettronica. La connessione al server avviene mediante l'utilizzo di socket.

Dopo aver caricato le impostazioni dal file *recvMail.conf*, l'applicazione tenterà di collegarsi al server POP che è stato indicato in queste.

Di seguito viene illustrato un esempio di questo file.



Stabilita la comunicazione inizia uno scambio di messaggi tra server POP e client.

Il primo passo effettuato è l'autenticazione dell'utente sul server. Di seguito viene riportato un esempio.

```
Server) +OK hello ...
Client) username
Server) +OK pass ...
Client) password
Server) +OK maildrop ...
```

Poi viene richiesto il numero di e-mail presenti sulla casella.

```
Client) LIST
Server) +OK numemail
Server) 1 ...
Server) 2 ...
Server) ...
Server) numemail ...
```

Successivamente l'applicazione legge le e-mail una ad una e cerca la stringa contenente i comandi, la quale avrà lo stesso formato di quella vista nella lettura degli SMS.

```
Client) RETR indicemail
Server) +OK dim ...
```

Infine l'email viene cancellata.

```
Client) DELE indicemail
Server) +OK
Server) Testo e-mail
```

Quando sono state lette tutte le email, l'applicazione richiede la chiusura della connessione.

```
Client) quit
Server) +OK ...
```

Come per la lettura degli SMS, se all'interno dei testi delle e-mail non ci sono stringhe nel formato valido, oppure se la password non corrisponde, verrà stampato un messaggio che ci avvisa che non sono stati trovati comandi, altrimenti il comando verrà inviato allo standard output (stdout).

decCMD.sh

Lo script decCMD.sh si occupa di decifrare il comando ricevuto via email/SMS ed eseguirlo. I comandi previsti e le azioni associate ad essi sono illustrate nella tabella pag. 16. Le azioni associate ai comandi sono costituite dalla combinazione delle funzioni illustrate a seguire:

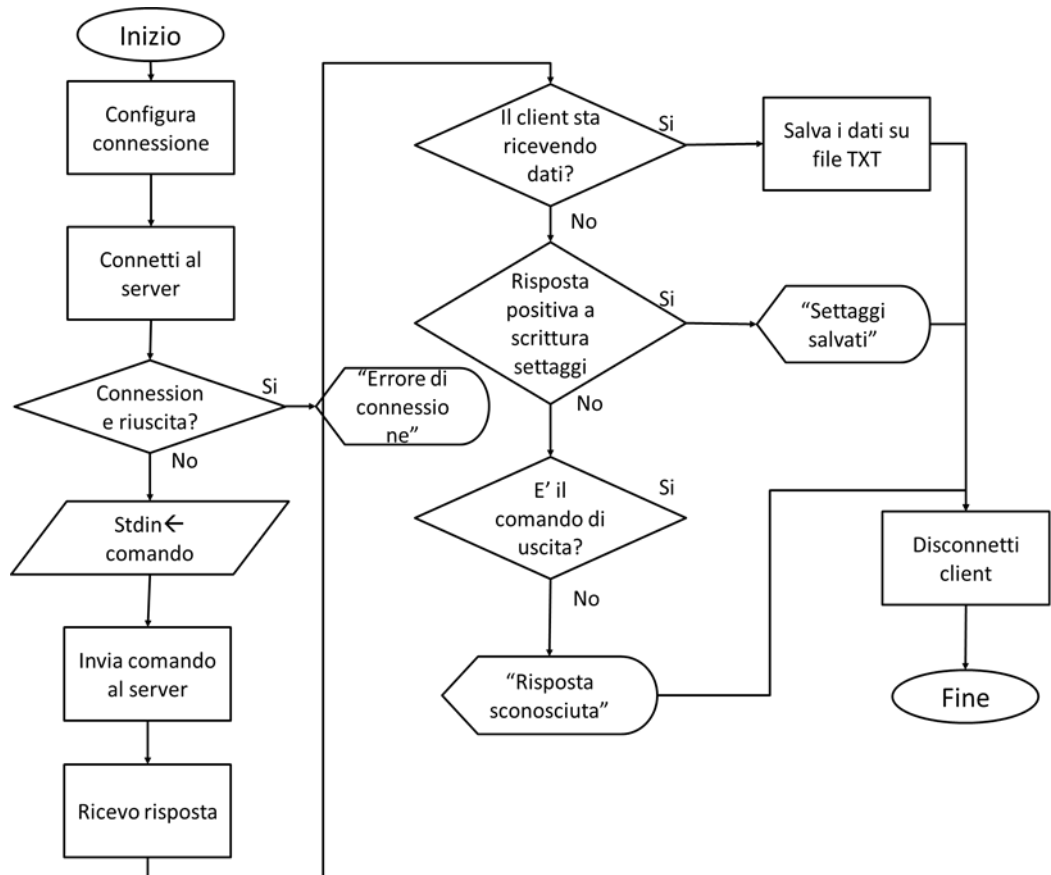
- **Accendi PC:** attraverso il software wakeonlan viene effettuato il "risveglio" del calcolatore remoto. Per "svegliare" il calcolatore remoto è stato utilizzato il *WakeOnLan (WOL)*. Il WOL è uno standard Ethernet che permette di avviare un computer in standby o spento da una postazione remota, conoscendone il MAC address. Sia la scheda madre sia la scheda di rete di tale computer, devono essere alimentate con una tensione di standby anche quando questo risulta spento. I pacchetti Ethernet che attivano la "sveglia" sono detti Magic Packet ("pacchetti magici"). Il Magic Packet è un frame multicast, composto da una prima parte di sincronizzazione costituita da sei bytes di valore FF esadecimale, seguiti dall'Indirizzo MAC del nodo bersaglio ripetuto 16 volte, per un totale di 96 bytes, e da un campo password opzionale che può essere assente, lungo 4 bytes o 6 bytes. L'applicazione adoperata è di semplice utilizzo, basta infatti indicare al comando *wol* l'indirizzo di broadcast e il MAC address del computer da risvegliare e i Magic Packet verranno inoltrati sulla rete Ethernet. Un esempio di comando sarà:

```
wakeonlan -i 192.168.0.255 -p 9 00:30:64:0E:8A:64
```

dove 192.168.0.255 è l'indirizzo di broadcast, e 00:30:64:0E:8A:64 è il MAC ADDRESS del computer da risvegliare.

- **Spegni PC:** questo comando sarà eseguito sull' embedded PC MXE, pertanto verrà inviato ad esso. L'inoltro del messaggio avviene tramite una comunicazione socket tra l'embedded PC e la SBC Raspberry. Il servizio sarà attivo sulla porta TCP 3000 dell'embedded PC.
- **Power ON:** il segnale di controllo del PTSK 240 viene settato "ALTO" e questo porta il relè allo stato di conduzione.
- **Power OFF:** il segnale di controllo del PTSK 240 viene settato "BASSO" e questo porta il relè allo stato di non conduzione.
- **Reboot SBC:** quando viene ricevuto questo comando, viene eseguito il comando reboot che riavvia la SBC.
- **OFF SBC:** spegne la SBC.
- **setc, getc, racq:** questi comandi devono essere eseguiti sul computer remoto, pertanto verranno inviati ad esso attraverso la medesima comunicazione utilizzata per lo spegnimento del PC. La descrizione di questi comandi viene effettuata di seguito nella descrizione del software lato Server della comunicazione socket.
- **Altrimenti:** se il comando ricevuto non è nessuno di quelli illustrati in precedenza, verrà segnalata la ricezione di un comando sconosciuto.

client



L'applicazione **client**, sviluppata in C, permette la comunicazione con il PC remoto mediante l'utilizzo dei socket. In particolare, avendo fissato l'indirizzo IP del server, questa applicazione tenterà di stabilire una connessione con l'indirizzo IP 192.168.0.1 sulla porta 3000. Se la connessione viene stabilita correttamente, questa applicazione permette di inviare dei comandi al server. Se la connessione non viene stabilita viene visualizzato un messaggio di errore. E' possibile ricevere tre tipi di risposte:

- Ricezione di dati a seguito di invio del comando racq. Vengono inviati al client i dati che questo ha richiesto.

- Risposta al comando setc. Viene inviata una risposta che indica che la scrittura dei settaggi è avvenuta correttamente.
- Risposta al comando exit. Dopo aver richiesto la chiusura dell'applicazione server, questa invita alla chiusura il client.
- Altrimenti. Il comando non è nessuno di quelli illustrati in precedenza, verrà quindi segnalata la ricezione di una risposta sconosciuta.

3.5.2.7 CONFIGURAZIONI SOFTWARE SBC

E' possibile cambiare i campi elencati a seguire per modificare la configurazione del software della SBC:

- Account e-Mail COMMAND IN
- Account e-Mail risposte
- APN connessione 3G
- PIN SIM 3G
- Password comandi
- MAC ADDRESS PC WAKE

Di seguito vengono elencati i percorsi dei file contenenti questi parametri e la posizione di questi all'interno di tali file, in modo tale da fornire indicazioni precise qualora fosse necessario un intervento.

- **/home/pi/connect3G**

line 7

```
if ! /home/pi/sendMail indirizzoMailRisposte IP /home/pi/netconf.txt
```

line 10

```
if ! /home/pi/sendMail indirizzoMailRisposte IP /home/pi/netconf.txt
```

- **/home/pi/remoteClient.sh**

line 10

```
while ! /home/pi/sendMail indirizzoMailRisposte Risposta_al_comando  
/home/pi/UT.tmp; do
```

line 159

```
while ! /home/pi/sendMail indirizzoMailRisposte Risposta_al_comando  
/home/pi/UT.tmp; do
```

- **/home/pi/decCmd.sh**

line 14

```
wakeonlan -i 192.168.0.255 -p 9 MACADDRESS_PCWAKE >>  
/tmp/OUT.tmp
```

line 28

```
wakeonlan -i 192.168.0.255 -p 9 MACADDRESS_PCWAKE >>  
/tmp/OUT.tmp
```

line 52

```
wakeonlan -i 192.168.0.255 -p 9 MACADDRESS_PCWAKE >>  
/tmp/OUT.tmp
```

line 77

```
wakeonlan -i 192.168.0.255 -p 9 MACADDRESS_PCWAKE >>  
/tmp/OUT.tmp
```

- **/home/pi/sendMail.conf**

line 1

indirizzoServerSMTP

line 2

portaServerSMTP

line 3

UsernameCodificatoBASE64eMailOUT

line 4

PasswordCodificatoBASE64eMailOUT

line 5

indirizzoeMailOUT

- **/home/pi/readSMS.conf**

line 1

PasswordComandi

- **/home/pi/recvMail.conf**

line 1

indirizzoServerPOP

line 2

portaServerPOP

line 3

UsernameMailIN

line 4

PasswordMailIN

line 5

PasswordComandi

- **/etc/sakis3G.conf**

line 5

CUSTOM_APN="APN_3G_CONNECTION"

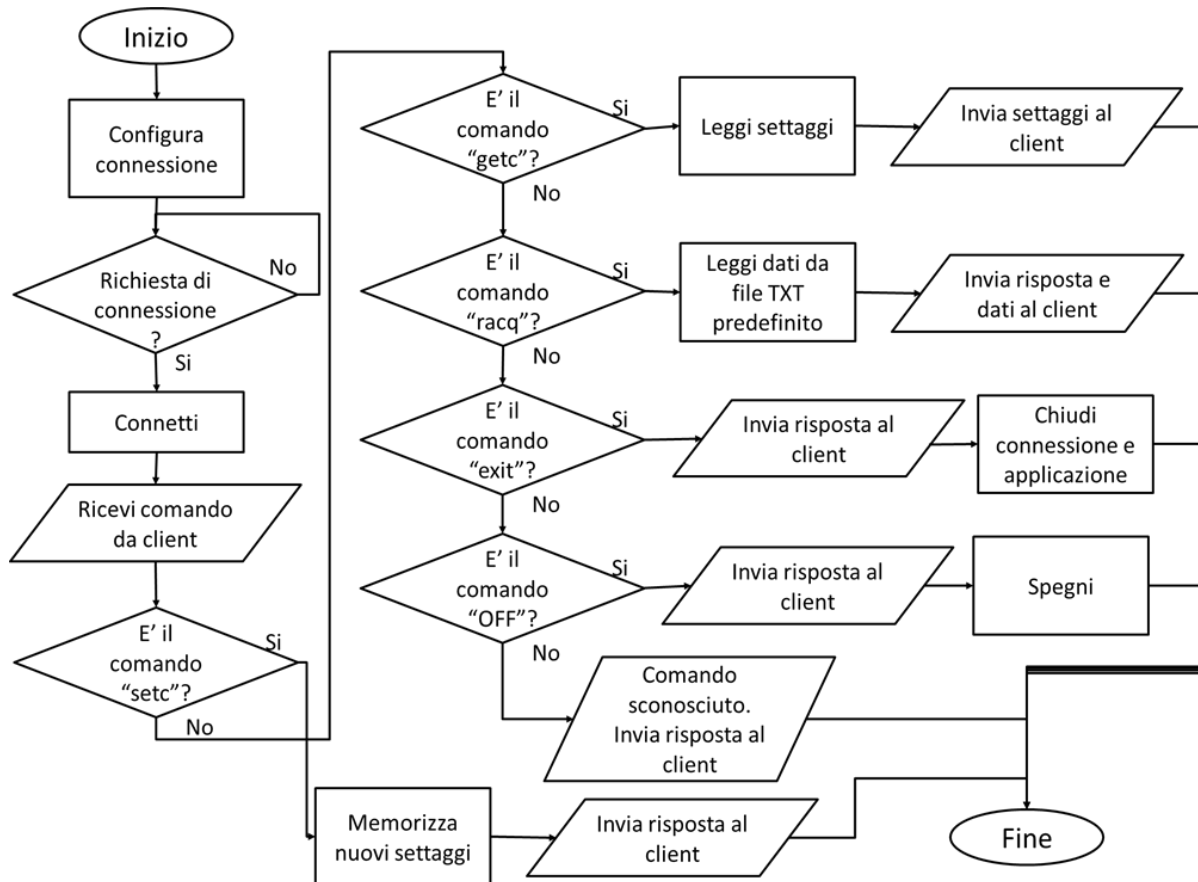
SIM_PIN="PIN_SIM"

3.5.2.8 SOFTWARE EMBEDDED PC

L'embedded PC MXE deve essere dotato dell'ambiente di sviluppo *National Instruments LabView 2011*. Tramite questo ambiente è stato creato un software lato server che permette di accettare e gestire richieste di connessione provenienti dalla rete Lan, alla quale questo è collegato mediante interfaccia Ethernet. Inoltre tale software può essere in grado di interfacciarsi con eventuale strumentazione elettronica di misura collegata all'embedded PC.

L'applicazione server è capace di riconoscere un certo numero di comandi inviati dal lato client. In particolare, oltre ai comandi di spegnimento e di trasferimento dati, prevede dei comandi per la lettura/scrittura di eventuali configurazioni. Di seguito vengono illustrati i comandi disponibili.

- exit** chiude l'applicazione server e invita alla chiusura al client.
- OFF** chiude l'applicazione server e attiva la procedura di spegnimento del PC dopo 15 sec.
- racq** invia il contenuto del file di testo che contiene eventuali dati.
- getc** restituisce i settaggi nel formato **-fCampo1-cCampo2-wCampo3-sCampo4**.
- setc** permette il settaggio dei parametri specificati con il formato sopra elencato.



Per un funzionamento ottimale del kit è consigliato un reboot della SBC Raspberry, manuale qualora fosse possibile altrimenti attraverso l'invio del relativo comando, ogni 24-36 ore.

3.5.2.9 TEST EFFETTUATI

L'intero sistema è stato testato a scopo di verificare il corretto funzionamento di quanto è stato implementato. E' stata verificata presso il Laboratorio del Gruppo di Elettronica la rispondenza tra i comandi inviati da remoto e le conseguenti azioni intraprese dal sistema di controllo. Particolare riguardo è stato rivolto alle condizioni di funzionamento critiche verificando che il sistema è in grado di superarle agevolmente. Il sistema è stato successivamente integrato nel radar in banda P.

