



ISTITUTO PROFESSIONALE DI STATO  
PER L'INDUSTRIA E ARTIGIANATO  
- MORETTO -

TESINA D' ESAME DI STATO

# Tesina di Maturità

## SMS MANAGER

BORRA ALESSANDRO  
DA SILVA CARLOS ANDRE  
CLASSE 5° BZ – SETTORE ELETTRONICO

ANNO SCOLASTICO 2004/05

## INDICE

Premessa .....	3
Introduzione .....	4
Il concetto trasmissione dei dati: .....	6
Organismi Internazionali di Standardizzazione .....	6
• Architettura del sistema .....	7
• Schema a blocchi .....	7
• Personal Computer(generale).....	7
Il sistema d'interfacciamento .....	8
• L'interfaccia Seriale RS232 .....	8
• Le unità di misura .....	9
• Com'è composto un segnale RS-232 .....	11
• Il bit di parità.....	12
• I parametri elettrici RS-232 .....	12
• Il MAX232 .....	13
• Cavi RS-232C .....	14
Il Modem (Pocket GSM) .....	15
• Linguaggio di comunicazione: .....	17
• Fase Pocket .....	18
• Invio SMS .....	19
• Lettura SMS .....	20
• Verifica Operatore .....	21
Il sistema GSM .....	23
• I servizi forniti dal gsm .....	25
• Modulo di Identificazione (SIM CARD).....	26
Il software .....	26
• Cronologia del prodotto BORLAND .....	27
• Il linguaggio Pascal.....	27
• Componenti.....	28
Spiegazione generale di SMS Manager .....	30
• Programma in lettura.....	31
• Pmach_status.....	31

• Macchina a stati finiti.....	34
• On show .....	35
• Listato delphi 1° programma .....	36 - 46
• Sintesi vocale degli SMS .....	47
• Listato delphi 2° programma .....	48 - 62
• Invio SMS .....	63
• Listato delphi 3° programma .....	65 - 69
• Er Finestra.....	69
Tesina di Inglese .....	71
• Telecommunication “Network” .....	72
Tesina di Storia .....	73
• Repubblica Armena.....	74
• Storia del genocidio .....	75
• La vita di Armin T Wergner .....	76
• Fotografie .....	77
• Il perché del genocidio.....	78
• Modalità dello sterminio .....	79
• L’Armenia attuale .....	81
• Il riconoscimento .....	81
• Riflessioni .....	82
Bibliografie del materiale storico.....	83
Bibliografia generale .....	84

## PREMESSA:

*Il nostro progetto nasce dalla voglia di sviluppare uno strumento che potesse aiutare a migliorare la vita del prossimo, con quest'idea abbiamo aderito alla proposta dei nostri insegnanti, di partecipare assieme ad altri alunni al Progetto ADA (Amici Diversamente Abili), nato da un'esperienza di lavoro volontario che ha permesso ad una persona diversamente abile di riacquistare importanti autonomie, nonostante una tetraplegia e gravi problemi di respirazione e parola, questo amico diversamente abile é oggi in grado di accendere le luci nella sua camera, di cambiare i canali del televisore, il computer, di scrivere e di comunicare attraverso la posta elettronica ed Internet. Nel 2003 gli insegnanti dell'IPSLA "Moretto" di Brescia che avevano partecipato all'esperienza hanno deciso, coinvolgendo alcuni studenti, di proseguire l'impegno cercando di offrire ad altre persone la stessa autonomia. Con questa filosofia ADA, che si propone di utilizzare l'elettronica e l'informatica per costruire soluzioni personalizzate, abbiamo sviluppato il programma "SMS Manager".*

## INTRODUZIONE:

La crescita della domanda di servizi radio mobili ha assunto in Italia un ritmo di crescita vertiginoso. Si ha la richiesta, da parte dell'individuo che si muove in un edificio all'esterno di esso, della possibilità di poter comunicare con altri individui, sia localmente che trasmettendo dati.

L'utilizzo di sistemi radio per la comunicazioni è relativamente antico, questa tecnologia è figlia di un'invenzione di Guglielmo Marconi(1895), e si sviluppò immediatamente dopo l'invenzione della radio. Inizialmente era utilizzato in ambito marittimo in modo da poter comunicare con stazioni fisse a terra, successivamente le radiocomunicazioni divennero elemento fondamentale per lo sviluppo della navigazione aerea, senza il quale lo sviluppo sarebbe stato certamente ritardato.

Gli aspetti da considerare per un significativo sviluppo anche terrestre erano molteplici, ad esempio di carattere tecnologico, organizzativo, economico.

### 1) Vincoli tecnologici:

Un sistema radiomobile è definito tale quando il collegamento tra due utenti è bidirezionale in aria. Vi è quindi una sostanziale differenza tra la radiodiffusione, che è monodirezionale (Broadcast) in tutto il sistema, e la radiocomunicazione che è bidirezionale per coppia utenti. In quest'ultimo caso, un singolo canale radio è utilizzato da una coppia di utenti per volta.

Dunque per fornire un servizio radiomobile ad un numero significativo di utenti sono necessari un numero rilevante di canali, cioè l'utilizzo di una quota importante dello spettro radioelettrico.

### 2) Vincoli organizzativi

La rete telefonica, cioè l'infrastruttura di base, veniva gestita da quattro enti : **Asst** (Azienda di Stato per i Servizi Telefonici ), **Sip** ( Società telefonica Italiana per l'esercizio delle Telecomunicazioni ), **Italcable e Telespazio**. Le ultime tre appartengono al gruppo **Iri** ( Istituto Ricostruzione Industriale) tramite la finanziaria **STET**. Ognuno di questi enti aveva competenze specifiche. **Asst** cura e gestisce i collegamenti internazionali relativi all'Europa e all'area del Mediterraneo. **SIP** curava e gestisce i collegamenti nazionali e gestisce il rapporto con l'utenza. **Italcable** curava e gestiva i collegamenti intercontinentali. **Telespazio** curava e gestiva i collegamenti via satellite.

### 3) Vincoli economici

Negli anni '70/'80 a causa dell'inflazione c'era stato un lentissimo ammodernamento delle infrastrutture. Dopo gli anni '80, grazie ad un clima politico favorevole ed ai maggiori investimenti nel settore, vi fu un maggiore sviluppo tecnologico. L'introduzione del sistema radio mobile era necessario per la risoluzione dei problemi del settore terziario, in quanto vi era la necessità di poter comunicare con il mondo ovunque ci si trovasse.

- **IL CONCETTO TRASMISSIONE DEI DATI:**

La trasmissione dei dati è l'insieme delle tecniche e delle problematiche connesse con il trasferimento a distanza di segnali informativi di tipo digitale tra elaboratori elettronici, utilizzando allo scopo le risorse, e in particolare i canali di comunicazione, messi a disposizione dai sistemi di telecomunicazione.

Negli ultimi anni, invece, la situazione è radicalmente cambiata per via dell'enorme sviluppo della microelettronica e della diffusione sempre maggiore di calcolatori in ogni ambito, sia aziendale sia domestico. I calcolatori sono sempre più utilizzati anche come veri e propri apparati per la comunicazione a distanza tra utenti una rete, come dimostra l'enorme successo avuto da Internet. Si sta così assistendo alla trasformazione degli elaboratori da puro strumento di calcolo ad apparato di telecomunicazione tramite il quale è possibile scambiare in modo digitale segnali di varie natura (voce, immagini, dati).

- **APPARATI INFORMATICI E DI TELECOMUNICAZIONE IN UN SISTEMA DI TRASMISSIONE:**

- Elaboratori:

Possono essere microprocessori, microcalcolatori o grossi calcolatori (su cui risiedono, per esempio le banche dati).

- Aparecchi di telefonia:

Possono essere telefono mobile, telefax.

- **ORGANISMI INTERNAZIONALI DI STANDARDIZZAZIONE:**

Esistono diversi organismi internazionali di standardizzazione che si occupano di trasmissione dati, i principali sono:

**ITU - (International Telecommunications Union);**

**ETSI – (European Telecommunications Standards Institute);**

**ISO - (International Organization for Standardization);**

**IEC – (International Electrotechnical Commission).**

**IEEE - (Institute of electrical and electronics engineers)**

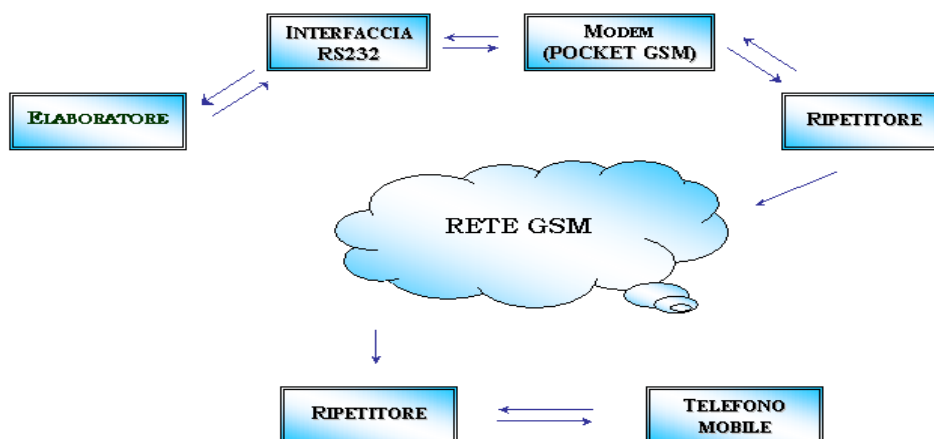
L'organismo di standardizzazione principale per le telecomunicazioni è l'ITU: un'organizzazione intergovernativa all'interno della quale cooperano sia enti pubblici sia privati al fine di consentire lo sviluppo delle telecomunicazioni.

Il nostro progetto può essere suddiviso nei seguenti blocchi:

- Elaboratore (Personal Computer)
- Interfaccia RS232
- Modem (Pocket GSM)
- Ripetitore
- Rete GSM
- Telefono Mobile

SCHEMA A BLOCCHI :

---



*Tesina d'Esame di Stato  
Anno 2004/05  
Ipsia Moretto Technology - Delphi LAB*

ELABORATORE :

---

## Il Personal Computer

Per descrivere un PC, può essere utile pensarlo in termini di sottosistemi che lo compongono. Questo rapido sommario dei suoi componenti può servire a mettere tutto in una corretta prospettiva:



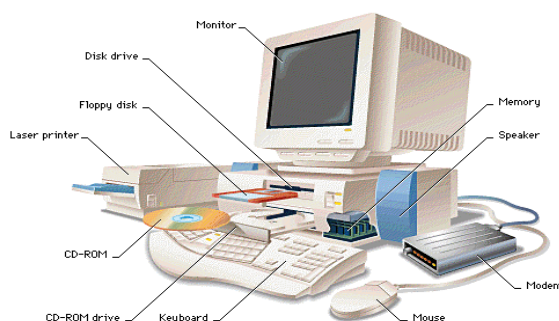
- (Aspetto esteriore) si vedono lo **schermo**, la **tastiera**, il **mouse** e l'**involucro del PC**. L'involucro contiene tutti i componenti essenziali per il PC e fornisce anche l'accesso alle unità CD-ROM e floppy nella parte anteriore e a varie porte

sul retro. L'involucro inoltre protegge l'utente dalle radiazioni elettromagnetiche prodotte dai componenti interni e consente al flusso d'aria canalizzato di ridurre i problemi di surriscaldamento.

- (Componenti interni) **CPU**: L'unità di elaborazione centrale (Central Processing Unit), o processore, che gestisce i programmi attivi sul PC. La velocità della CPU, combinata con il suo design interno, determina la velocità e l'efficienza con cui possono essere elaborati i dati e i codici dei programmi.
- **Memoria di sistema**: Chiamata anche memoria ad accesso casuale (Random Access Memory), o RAM, la memoria di sistema immagazzina i codici e i dati che il PC sta utilizzando. Più memoria si ha, migliori sono le prestazioni.
- **Disco rigido**: È il dispositivo per l'immagazzinamento permanente dei dati sul PC. I programmi e i dati contenuti nel disco rigido vengono trasferiti nella RAM di sistema per essere usati. I dischi rigidi odierni vantano capacità di 20 GB e oltre.
- **Unità CD-ROM**: L'onnipresente supporto rimovibile di grande capacità. Le unità CD-ROM permettono agli utenti del PC di accedere ai contenuti multimediali, di installare applicazioni di grande dimensioni e di lavorare con enormi data base e con altre risorse. Questa tecnologia sta lasciando il posto alle unità DVD-ROM, che offrono capienze ancora più grandi per i dati (sull'ordine dei 17 GB).
- **Scheda grafica**: Tutto ciò che si vede sullo schermo passa attraverso la scheda grafica. Questo componente hardware accelera la visualizzazione delle immagini, sia in due sia in tre dimensioni, e delle riproduzioni dei video. Una buona scheda grafica è fondamentale per avere una buona visione e per prestazioni veloci.
- **Scheda audio**: Tutto ciò che va oltre gli elementari bip e che è emesso dall'altoparlante del PC passa attraverso il sottosistema audio. Le schede audio permettono di riprodurre i suoni digitali attraverso gli altoparlanti, nonché di registrare suoni analogici in formato digitale.
- **Modem**: Consente al PC di "parlare" con Internet, con i fax e con altri PC, usando una normale linea telefonica. Può essere sia una scheda aggiunta internamente sia un dispositivo esterno.
- **Scheda di interfaccia con la rete**: Questa scheda supplementare serve a connettere il PC a una rete locale.

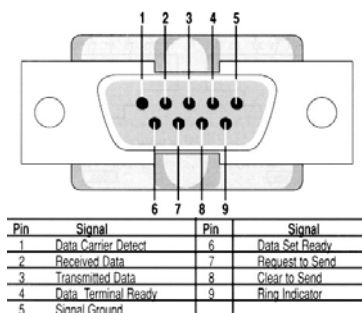
## IL SISTEMA D'INTERFACCIAMENTO:

I sistemi di interfacciamento, vengono utilizzati nei PC per connettere il sistema di elaborazione dati con i dispositivi periferici esterni (tastiera, mouse, video, stampante, plotter, unità HD, FD, CDROM, modem etc). Alcuni sistemi di interfacciamento sono essenziali per il funzionamento del sistema stesso (interfaccia video, interfaccia tastiera), altri sono opzionali. Fra le modalità di trasferimento dati distinguiamo la “modalità parallelo” attuata ad esempio nel caso dell’interfaccia di una stampante e interfacciamento seriale (verso un modem o un plotter o una stampante seriale).



## L'INTERFACCIA SERIALE RS232

L'interfaccia seriale è definita da uno standard noto come RS-232. L'interfaccia R2-232 stabilisce le modalità di comunicazione (a livello fisico) fra due dispositivi: il DTE (Data Terminal Equipment) e il DCE (Data Circuit Equipment). Il DTE è generalmente un computer, il DCE può essere un modem, un computer o altro dispositivo compatibile con l'interfaccia. Nel computer sono presenti due connettori (9/25 pin) collegati a due porte RS-232 (COM1 e COM2). Nella figura è riportato il pinout del connettore a 9 pin.



(la fig. sopra riportata rappresenta la configurazione di ogni pin)

## **L'interfaccia RS-232 utilizza un protocollo seriale di tipo asincrono.**

Interfaccia Seriale: significa che i bit che costituiscono l'informazione sono trasmessi uno alla volta su di un solo "filo". Questo termine è in genere contrapposto a "parallelo": in questo caso i dati sono trasmessi contemporaneamente su più fili, per esempio 8, 16 o 32.

Parlando astrattamente si potrebbe pensare che la trasmissione seriale sia intrinsecamente più lenta di quella parallela (su di un filo possono passare meno informazioni che su 16). In realtà questo non è vero in assoluto, soprattutto a causa della difficoltà di controllare lo skew (disallineamento temporale tra i vari segnali) dei molti trasmettitori in un bus parallelo, e dipende dalle tecnologie adottate: per esempio in una fibra ottica, in un cavo ethernet, USB o FireWire (tutti standard seriali) le informazioni transitano ad una velocità paragonabile a quella di un bus PCI a 32 fili. In questa nota applicativa si parlerà solo di interfacce seriali "lente" cioè gestibili da PC e microcontrollori "normali".

**Asincrono** significa che i dati sono trasmessi senza l'aggiunta di un segnale di clock, cioè di un segnale comune che permette di sincronizzare la trasmissione con la ricezione; ovviamente sia il trasmettitore che il ricevitore devono comunque essere dotati di un clock locale per poter interpretare i dati. La sincronizzazione dei due clock è necessaria ed è fatta in corrispondenza della prima transizione sulla linea dei dati.

### Le unità di misura

---

Le unità di misura della velocità di trasmissione sono essenzialmente due: il *baud* ed il *bit per secondo* (bps o, meno spesso, b/s), spesso trattate erroneamente come sinonimi.

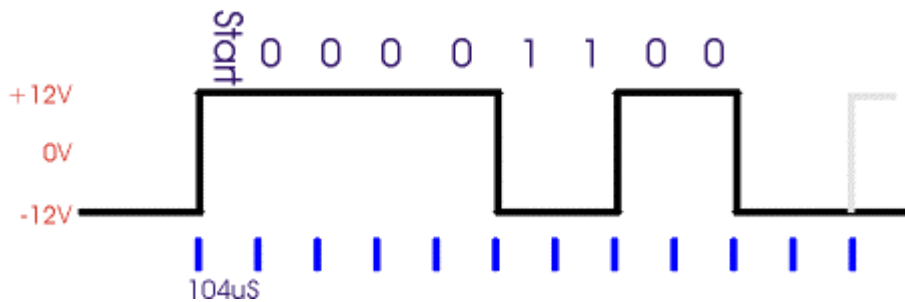
Il *baud rate* indica il numero di transizioni al secondo che avvengono sulla linea; il *bps* indica, come dice il nome, quanti bit al secondo sono trasmessi lungo la linea.

Nel caso di trasmissione binaria (cioè è presente un livello alto ed uno basso) le due cose ovviamente coincidono numericamente, da cui la *parziale* equivalenza dei due termini. Nel caso di trasmissioni a più livelli, invece, è possibile trasmettere con una sola transizione più bit: se per esempio posso trasmettere otto diversi valori di tensione tra 0 ed 7 volt, con un solo valore di tensione invio tre bit (0V = 000, 1V = 001, 2V = 010...) ed in questo caso una trasmissione a 1000 baud equivale ad una a 3000 bps.

In questa nota parlerò solo di segnali binari trasmessi secondo lo standard RS232: quindi il baud rate, pari all'inverso della durata di un bit, coincidente numericamente con il bps.

La cosa più semplice per descrivere un segnale RS232 è partire con un esempio.

Nell'immagine che segue è visualizzato, in modo idealizzato, cosa appare collegando un oscilloscopio ad un filo su cui transita un segnale RS-232 a 9600 bps del tipo 8n2 rappresentante il valore binario 00110000.



L'ampiezza del segnale è caratterizzata da un valore "alto" pari a circa +12V ed un valore "basso" pari a -12V. Da notare che, nello standard RS-232 un segnale alto rappresenta lo zero logico ed uno basso un uno, come indicato nel disegno e rovesciato rispetto al comune pensare.

A volte un segnale alto (+12V, cioè uno zero logico) è indicato come space ed uno basso (-12V, uno logico) come mark.

Tutte le transizioni appaiono in corrispondenza di multipli di 104us (pari ad 1/9600 cioè ciascun bit dura esattamente l'inverso del baud rate).

La linea si trova inizialmente nello stato di riposo, bassa (nessun dato in transito); la prima transizione da basso in alto indica l'inizio della trasmissione (inizia il "bit di start", lungo esattamente 104us). Segue il bit meno significativo (LSB), dopo altri 104 uS il secondo bit, e così via, per otto volte, fino al bit più significativo (MSB). Da notare che il byte è trasmesso "al contrario", cioè va letto da destra verso sinistra. Segue infine un periodo di riposo della linea di *almeno* 208us, cioè due bit di stop e quindi (eventualmente) inizia un nuovo pacchetto di bit.

**In genere il formato del pacchetto trasmesso è indicato da una sigla composta da numeri e cifre, per esempio 8N1 e 7E2:**

- La prima cifra indica quanti bit di dati sono trasmessi (nei due esempi rispettivamente 8 e 7)
- La prima lettera il tipo di parità (rispettivamente nessuna ed even-parity, cioè parità pari)
- La seconda cifra il numero di bit di stop (rispettivamente 1 e 2)

Nei normali PC le cosiddette interfacce seriali RS-232 arrivano in genere almeno a 115Kbps o anche più: pur essendo formalmente al di fuori di ogni standard ufficiale non si hanno particolari problemi di interconnessione.

Ovviamente sia trasmettitore che ricevitore devono accordarsi sul modo di trasmettere i dati prima di iniziare la trasmissione.

E' importante garantire il rigoroso rispetto della durata dei singoli bit: infatti non è presente alcun segnale di clock comune a trasmettitore e ricevitore e l'unico elemento di sincronizzazione è dato dal fronte di salita del bit di start. Come linea guida occorre considerare che il campionamento in ricezione è effettuato di norma al centro di ciascun bit: l'errore massimo ammesso è quindi, teoricamente, pari alla durata di mezzo bit (circa il 5% della frequenza di clock, considerando che anche il decimo bit deve essere correttamente sincronizzato). Naturalmente questo limite non tiene conto della possibile difficoltà di riconoscere con precisione il fronte del bit di start (soprattutto su grandi distanze ed in ambiente rumoroso) e della presenza di interferenze intersimboliche tra bit adiacenti: per questo spesso si consiglia caldamente di usare un clock con una precisione migliore dell'1% imponendo di fatto l'uso di oscillatori a quarzo. Si potrebbe anche ipotizzare un meccanismo che tenta di estrarre il clock dai fronti intermedi ma si tratta nel caso specifico di un lavoro poco utile, visto che la lunghezza del pacchetto è piuttosto breve.

## Il bit di parità

---

Oltre ai bit dei dati (in numero variabile tra 5 ed 9) viene inserito un bit di parità (opzionale) per verificare la correttezza del dato ricevuto. Esistono cinque tipi di parità:

- None: nessun tipo di parità, cioè nessun bit aggiunto
- Pari (even): il numero di mark (incluso il bit di parità) è sempre pari
- Dispari (odd): il numero di mark (incluso il bit di parità) è sempre dispari
- Mark: il bit di parità vale sempre [mark](#)
- Space: il bit di parità vale sempre [space](#)

## I PARAMETRI ELETTRICI RS-232

---

La tensione di uscita ad un trasmettitore RS232 deve essere compresa in valore assoluto tra 5V e 25V (quest'ultimo valore ridotto a 13V in alcune revisioni dello standard). A volte le tensioni in uscita sono intenzionalmente diminuite a +/- 6V anziché 12V per permettere minori emissioni EMC, peraltro sempre critiche, e favorire maggiori velocità di trasmissione.

Il ricevitore deve funzionare correttamente con tensioni di ingresso comprese, sempre in modulo, tra i 3V ed i 25V. Molti ricevitori commerciali considerano semplicemente una tensione di soglia al

valore di +2V (sopra viene riconosciuto un segnale alto, sotto uno basso) anche se ciò non è pienamente aderente alla norme. E' però utile per effettuare una trasmissione "rs232" con livelli TTL.

L'impedenza di uscita del trasmettitore deve in ogni situazione essere maggiore di 300 ohm; l'impedenza di ingresso deve essere compresa tra i 3 ed i 7 kohm, anche a dispositivo spento. La corrente prelevabile in uscita mantenendo i corretti valori logici deve essere di almeno di 1.6 mA (potrebbe però essere maggiore, anche di un ordine di grandezza) e nel caso di corto circuito deve essere minore di 100mA.

Infine lo slew-rate (cioè la pendenza del grafico del segnale nel passare da 1 a 0 o viceversa) deve essere minore di 30V/us per evitare eccessive emissioni elettromagnetiche.

## IL MAX232

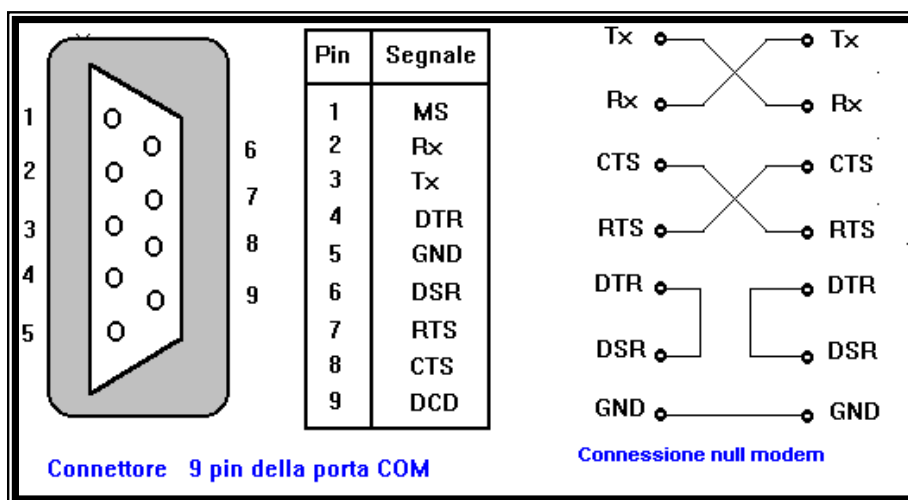
Il MAX232 (ed integrati simili, fatti da un po' tutti i produttori di semiconduttori) è un circuito integrato che permette il collegamento tra logica TTL o CMOS a 5V e le tensioni RS-232, partendo solo da un'alimentazione a 5V.

Sono disponibili anche integrati che richiedono un'alimentazione di solo 3.3V (p.e. il MAX3232).

La sezione ricevente del MAX232 è costituita da due porte invertenti che accettano in ingresso una tensione di +/- 12V (o altra tensione compatibile allo standard RS232) ed in uscita hanno un segnale TTL compatibile.

La sezione trasmittente ha due driver invertenti con in ingresso TTL compatibile e capaci di erogare a vuoto una tensione di poco meno di +/- 10V, compatibile con lo standard RS232.

### LA PIEDINATURA DEL CONNETTORE RS-232 DEL PC



Si tratta dei cavi utilizzati per connettere un'unità periferica a una porta seriale. A seconda dei componenti da connettere tra loro, si parla di DTE (*Data terminal equipment*) e DCE (*Data communications equipment*). L'elaboratore è sempre un DTE, il modem è un'unità DCE, mentre una stampante o un terminale può essere un DTE.

Quando si connettono due unità eterogenee, come un elaboratore con un modem, si utilizza un cavo seriale composto da un connettore DB-25 maschio, da collegare all'unità periferica DCE, e da un connettore DB-25 o DB-9 femmina, da collegare alla porta seriale dell'elaboratore (DTE). Con questo tipo di cavo, tutti i segnali di un capo sono connessi con gli stessi segnali dell'altro.

Cavo seriale RS-232C standard (DTE-DCE)

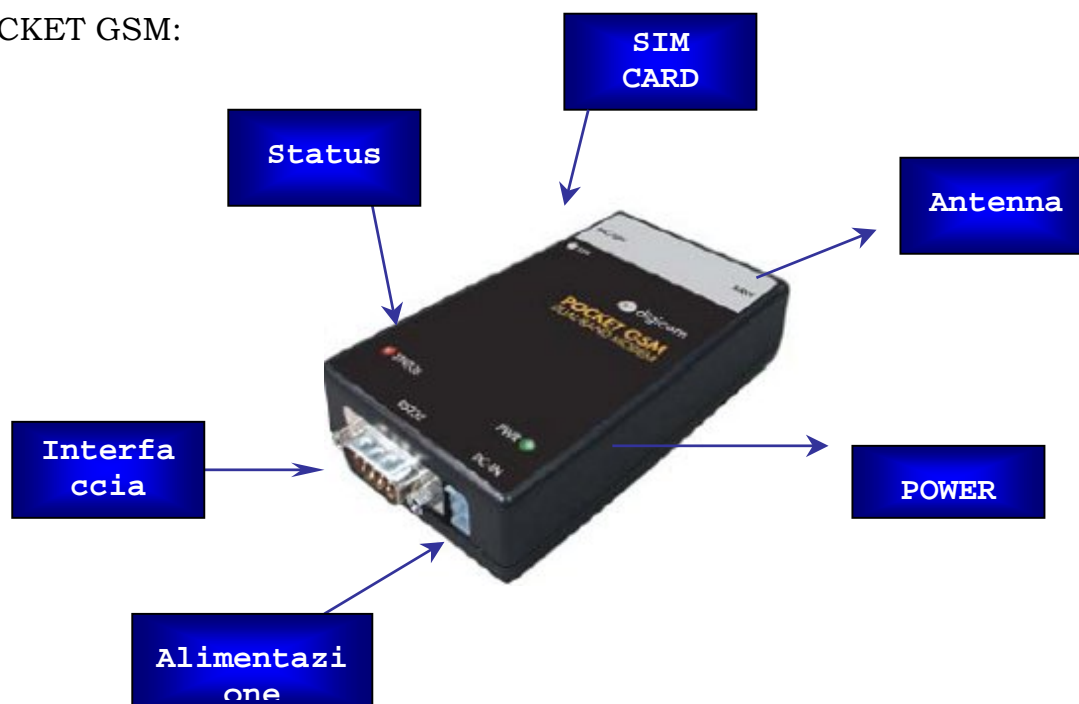
<b>Segnale</b>	<b>DTE (elaboratore)</b>	<b>DTE (elaboratore)</b>	<b>DCE (modem)</b>
	<b>DB-25</b>	<b>DB-9</b>	<b>DB-25</b>
<b>TD Transmit data</b>	2	3	2
<b>RD Receive data</b>	3	2	3
<b>RTS Request to send</b>	4	7	4
<b>CTS Clear to send</b>	5	8	5
<b>DSR Data set ready</b>	6	6	6
<b>Massa dei segnali</b>	7	5	7
<b>DCD Data carrier detect</b>	8	1	8
<b>DTR Data terminal ready</b>	20	4	20
<b>RI Ring indicator</b>	22	9	22

Un cavo Null-modem, per la connessione tra due elaboratori (o comunque due unità DTE) attraverso la porta seriale, può essere realizzato utilizzando due connettori DB-25 femmina, oppure DB-9 femmina, oppure un DB-25 e un DB-9 femmina. Se si intende utilizzare un controllo di flusso software, ovvero XON/XOFF, sono sufficienti tre fili, mentre per un controllo di flusso hardware, ovvero RTS/CTS, sono necessari sette fili. Se il cavo ha una schermatura metallica, questa può essere connessa alla parte metallica di uno solo dei due connettori.

## Il Modem (Pocket GSM)

Un modem (acronimo di modulatore/demodulatore) converte il segnale dati digitale del PC nel segnale analogico usato dal sistema telefonico: il modem può essere installato all'interno del PC in uno slot di espansione, oppure connesso al PC esternamente tramite una porta seriale o USB.

POCKET GSM:



*Schema generale*

- Modem Dual-Band GSM 900/1800 MHz
- Trasmissione dati fino a 9.6Kbps V.32 e V.110
- Trasmissione e Ricezione SMS
- Aggiornabile via Flash
- Dual-Band GSM 900/1800 MHz
- Data Transmission up to 9.6Kbps V.32 and V.110
- Send & Receive SMS
- Flash Upgradable

**Pocket GSM è un modem GSM Dual-band** (900-1800Mhz) viene utilizzato per applicazioni per la trasmissione dei dati via etere utilizzando la rete GSM come per esempio l'invio di piccoli messaggi di testo SMS(Short message service).

E' stato disegnato per operare con le reti di tutti gli operatori GSM sia in modo diretto che in roaming. Pocket GSM è conforme alla Classe 4 (900 Mhz) e Classe 1 (1800 Mhz).

- **Alimentazione**

L'alimentazione del modem deve essere compresa tra i +12 e i 24Vcc  $\pm 20\%$  con negativo a massa.

- Collegamento antenna esterna

**Per collegare l'antenna esterna, procedete nel modo seguente:**

1. Assicuratevi che il Pocket GSM sia spento, scollegando il connettore di alimentazione
2. Collegate il connettore proveniente **dall'antenna esterna** avvitandolo in senso orario esercitando una forza normale. In caso di difficoltà non forzate assolutamente il connettore ma verificatene il corretto posizionamento. Posizionare l'antenna, avendo cura di verificare la presenza di un buon segnale dell'operatore GSM.



E' possibile verificare il segnale GSM in tre modalità:

- **Attraverso un telefono cellulare**

Inserite la SIM che andrà nel Pocket GSM in un telefono cellulare e verificate il punto in cui il segnale è maggiore.

- **Attraverso il led Status** (vedere: Indicatori led)

Alimentate il modem e verificate se il LED status. Il LED rosso fisso indica che il modem è registrato alla rete GSM, in questo modo si può verificare se il segnale GSM è presente, ma non la quantità del segnale.

- **Attraverso il comando AT+CSQ** (vedere il paragrafo: Verifica del livello del segnale)

Attraverso questo comando si ha la possibilità di verificare con maggiore precisione la zona migliore in cui installare l'antenna.

Alimentate il Pocket GSM, collegatelo tramite il cavo seriale ad un Personal computer e tramite un emulatore di terminale (nel nostro caso la suite Dephi) inviate al modem il comando AT+CSQ. Il modem risponderà con +CSQ:x,y; dove "x" indica la qualità del segnale e "y" il BER (Bit Error Rate). la qualità del segnale "x" potrà avere un valore compreso tra 0 e 31; i valori più alti indicano un segnale migliore.

Da 0 a 9 = Scarso/insufficiente
Da 10 a 15 = Buono
Da 16 a 31 = Ottimo

Il Pocket GSM è dotato di una serie di comandi AT che permettono la programmazione del dispositivo e l'esecuzione di tutte le funzionalità supportate dal Pocket GSM. Di seguito verranno descritte alcune delle applicazioni.

AT [ *comando . . .* ]

I comandi dei modem compatibili Hayes iniziano quasi sempre per «AT» seguito da una serie eventuale di codici di comando alfanumerici e quindi da un codice di ritorno a carrello (<CR>).

Per esempio:

- ATDP chiamata a impulsi (telefono decadico);
- ATDT chiamata a toni (telefono multifrequenza).

I comandi di base iniziano con una lettera alfabetica; a questi sono stati aggiunti nel tempo dei comandi estesi che possono iniziare con una e-commerciale (&), un simbolo di percentuale (%), una barra obliqua inversa (\) e altri simboli ancora. Quando si fa riferimento a comandi estesi, è difficile stabilire quale sia lo standard; qui si vogliono elencare solo i comandi di base e quelli estesi più comuni e quindi più importanti.

Alcuni comandi speciali non fanno uso del solito prefisso di comando AT. Sono pochi e piuttosto importanti.

I comandi seguenti richiedono il prefisso AT e sono seguiti dal carattere di ritorno a carrello (<CR>). I comandi prefissati da AT possono essere più o meno complessi e lunghi di conseguenza; questa lunghezza ha un limite che varia da modem a modem. In generale, quando possibile, è opportuno suddividere questi comandi se sono troppo lunghi.

La maggior parte dei casi, i comandi AT sono formati da una sigla iniziale che definisce il tipo di comando e sono seguiti da un parametro numerico. Per esempio, ATH0 serve a chiudere la linea telefonica. Questi comandi possono essere composti senza il parametro finale (cioè senza il numero), quando si vuole fare riferimento allo zero. Quindi, ATH è esattamente uguale a ATH0.

I comandi AT possono contenere spazi, per facilitare la lettura umana. Resta comunque valido il problema del limite massimo alla loro lunghezza, che in tal modo deve tenere conto anche degli spazi aggiuntivi (ammesso che il modem non ne tenga conto esplicitamente).

- a) Gestione PIN
- b) Trasferimento Dati
- c) Gestione SMS
- d) Indicazione nuovi messaggi SMS
- e) Verifica operatore
- f) Livello del segnale

#### A) GESTIONE PIN

Solitamente le SIM richiedono l'inserimento del codice PIN. Se nell'applicazione a cui è destinato il Pocket GSM non è richiesto l'inserimento del PIN ad ogni accensione è possibile disabilitare sulla SIM la richiesta utilizzando un telefono cellulare. Attraverso il comando AT+CPIN è comunque possibile inserire il PIN.

Esempio:

Comando: at+cpin? (per verificare lo stato del PIN)

Risposta: +CPIN: SIM PIN (è richiesto l'inserimento del PIN)

*OK*

Comando: at+cpin="4321" (inserimento PIN)

Risposta: OK

Comando: at+cpin?

Risposta: +CPIN: READY (la SIM è pronta)

*OK*

B) Questo tipo di comando non'è stato utilizzato.

#### C) GESTIONE SMS

Il Pocket GSM gestisce l'invio e la ricezione degli SMS solo in formato testo.

## Invio SMS

Di seguito verranno indicati una serie di comandi necessari per la gestione dell'invio SMS.

La prima operazione da effettuare è inserire il numero del centro servizi.

1) Impostazione/lettura del Centro servizi:

Comando: AT+CSCA="+393205858500"<INVIO> (es. Centro Servizi WIND)

Nota: Il numero del Centro Servizi deve essere delimitato dal carattere “.

Comando: AT+CSCA? (lettura del centro servizi memorizzato)

Risposta: +CSCA: "+393205858500",145

OK

2) Definizione formato:

Comando: AT+CMGF=1 <INVIO> (Text Mode)

Risposta: OK

3) Salvataggio impostazioni:

Comando: AT+SCRT<INVIO>

Risposta: OK

4) Invio messaggio:

Comando: AT+CMGS="+393217654321"<INVIO>

- Prova invio SMS in formato testo <ctrl z>

Risposta: OK

Dopo aver digitato il messaggio che si vuole inviare, premere <ctrl z> (Hex =1°). In questo esempio viene inviato il messaggio “Prova invio SMS in formato testo” al numero 3217654321.

## Letture SMS

Comando: AT+CMGL="ALL"<INVIO>

Risposta: +CMGL: 1,"REC UNREAD","+393312345678", "01/06/06,11:36:18+04"

PROVA CIAO

+CMGL: 2,"REC UNREAD","+393312345678", "01/06/06,15:43:10+04"

PROVA RICEZIONE SMS

OK

### AT+CMGL Lista dei messaggi

Questo comando è utilizzato per visualizzare la lista dei messaggi SMS ricevuti dalla rete e/o memorizzati nella rubrica interna del modem.

#### Comando Possibili risposte Azione

AT+CMGL=<stat> OK/ERROR Visualizza i messaggi SMS specificati da <stat>

AT+CMGL=? +CMGL: (lista di <stat>) Visualizza i valori supportati

<stat>

"ALL" Visualizza tutti i messaggi (Memorizzati, letti o non letti)

"REC UNREAD" Visualizza solo i messaggi ricevuti ma non letti

"REC READ" Visualizza solo i messaggi ricevuti e letti

"STO UNSENT" Visualizza i messaggi memorizzati nella rubrica e non spediti

"STO SENT" Visualizza i messaggi memorizzati nella rubrica e spediti

#### Cancellazione SMS

Per cancellare il secondo messaggio (PROVA RICEZIONE SMS) il comando da dare è:

Comando: AT+CMGD=2

Risposta: OK

#### D)INDICAZIONE NUOVI MESSAGGI SMS

Tramite il comando AT+CNMI è possibile programmare il Pocket GSM per inviare verso il DTE l'informazione riguardo l'arrivo di un nuovo SMS.

Esempio:

Comando: AT+CNMI=1,1<INVIO>

Risposta: OK

Il Pocket GSM all'arrivo di un SMS invierà verso il DTE (p.e. PC) il seguente messaggio:

+CMTI: "SM",2 ( il 2 indica la locazione di memoria in cui viene memorizzato SMS)

#### E) VERIFICA OPERATORE

Attraverso il comando AT+COPS è possibile verificare se il Pocket GSM è registrato alla rete GSM ed eventualmente con quale operatore.

Esempi:

*Comando: AT+COPS?*

Risposta: +COPS: 0 (In assenza di campo o se la SIM non è presente)

OK

Comando: AT+COPS?

Risposta: AT+COPS? (Se il Pocket GSM è registrato alla rete GSM)

+COPS: 0,2,"22288"

OK

**Nota: "22288" corrisponde all'operatore GSM utilizzato.**

**I primi tre numeri identificano lo stato (222=ITALIA), mentre gli ultimi due identificano l'operatore (01=TIM, 10=Omnitel, 88=Wind, 98=BLU...).**

Inserendo il carattere = alla sintassi del comando +COPS (+COPS=?) è possibile verificare la lista degli operatori disponibili, e la corrispondenza con il codice ricevuto dal comando AT+COPS?.

Comando: AT+COPS=? (attendere alcuni secondi per la risposta)

Risposta: +COPS:(2,"I WIND","I WIND","22288")(1,"IT TIM","TIM","22201")(1,"I-OMNITEL","OM NI","22210"),(0-4), (0-2)

#### F) VERIFICA LIVELLO DEL SEGNALE

Per verificare il livello del segnale è possibile utilizzare il comando AT+CSQ

Comando: AT+CSQ

Risposta: +CSQ: 15,0 (15 corrisponde a -83dBm)

OK

Il valore di CSQ può essere compreso tra 0 e 31, dove 0 = -113 dBm e 31 = -51dBm.

Ogni unità corrisponde a 2 dBm.

Da 0 a 9 = Scarso/insufficiente

Da 10 a 15 = Buono

Da 16 a 31 = Ottimo

G) Settaggio velocità modem:

AT+CBST Selezione della velocità (verificare la lista dei valori supportati)

Questo comando permette di selezionare la velocità e le modalità della chiamata.

AT+CBST=<s>,<m>,<p> Sintassi del comando

AT+CBST? Visualizza i valori correnti

AT+CBST=? Visualizza i valori supportati

<s> Velocità

Valore Descrizione Standard	
0	Multistandard
1	300 bps
2	1200 bps
4	2400 bps
6	4800 bps
7	9600 bps
66	1200 bps
68	2400 bps
70	4800 bps
71	9600 bps
<m>	Modo

Breve storia del sistema, che viene utilizzato ancora tutt'ora :

Nei primi anni '80 vi fu una rapida crescita dei sistemi cellulari analogici in Europa. Ogni paese sviluppò però il proprio sistema, incompatibile con ogni altro sia in termini di software che di hardware. Il terminale mobile era limitato ad operare entro i confini nazionali.

La definizione di uno standard pan-europeo avrebbe consentito di operare in regime di concorrenza, per effetto della standardizzazione delle interfacce e delle funzioni che consente ai gestori di utilizzare impianti forniti da diversi costruttori. Si sarebbe potuto: aprire un vasto mercato in grado di permettere significative economie di scala nella produzione di terminali e apparati con conseguente diminuzione dei loro costi, e creare un servizio internazionale privo di confini.

Nel 1982 la *Conférence Européenne des Postes et des Télécommunications* CEPT creò, su proposta, un gruppo di studio *Groupe Spécial Mobile* (GSM) con lo scopo di studiare e sviluppare un sistema radiomobile cellulare paneuropeo comune a tutti i paesi dell'Europa occidentale.

Il sistema proposto doveva rispettare dei precisi criteri:

- Assicurare una buona qualità audio della conversazione
- Bassi costi per i terminali e per la gestione del servizio
- Supporto per il Roaming internazionale
- Supporto per terminali palmari
- Supporto per un ampio ventaglio di nuovi servizi
- Compatibilità con il sistema digitale ISDN
- Garantire un eccellente grado di sicurezza e riservatezza nelle comunicazioni.

Tre anni, dal 1982 al 1985 furono dedicati alla scelta tra la tecnica analogica e quella numerica. La decisione finale fu quella di adottare la seconda. Si scelse perciò una tecnologia digitale non ancora testata, in netta antitesi con la tecnologia dei già sperimentati sistemi cellulari analogici come AMPS e TACS.

A favore della tecnologia digitale, la rapida evoluzione tecnologica dei settori dell'elaborazione numerica dei segnali e l'integrazione dei componenti elettronici per effetto della disponibilità dei circuiti integrati VLSI.

Un sistema cellulare basato su sistema numerico offre numerosi vantaggi:

- consente di utilizzare una frequenza per servire più utenti, tramite l'utilizzo di tecniche TDM (Time Division Multiplexing),
- ha una capacità maggiore sia per quanto sopra detto, sia perchè i sistemi digitali sono meno sensibili a rumore ed interferenze e quindi consentono di ridurre le dimensioni delle celle, aumentando il numero di utenti che possono essere serviti contemporaneamente,
- consente alto grado di riservatezza, in quanto le informazioni trasmesse sulla tratta radio possono essere cifrate direttamente dall'apparato utente,
- consente elevato grado di sicurezza: l'identità dell'apparato che chiede l'accesso alla rete può essere controllata tramite l'applicazione di un opportuno algoritmo e di una chiave di autenticazione segreta,
- consente di effettuare trasmissioni dati (il segnale vocale stesso viene digitalizzato e poi trasmesso).

Nel 1987 superati i problemi tecnici e politici affrontati per uniformare i diversi punti di vista dei paesi coinvolti e dei numerosi studiosi che portavano avanti progetti e sperimentazioni, si arrivò alla stesura di un accordo *Memorandum of Understanding* MoU per l'introduzione coordinata del sistema GSM. Fu indicata come apertura del servizio il primo luglio 1991.

Nel 1989, la responsabilità del progetto GSM venne trasferita alla *European Telecommunication Standards Institute*, specificatamente a un Comitato Tecnico di ETSI. In quella sede venne ridefinito l'acronimo GSM come *Global System for Mobile Communications*. Il Comitato Tecnico ha elaborato normative, standard e specifiche tecniche descritte in dodici serie di raccomandazioni. La prima parte delle specifiche venne pubblicata nel 1990 (PHASE 1).

Dopo la fase iniziale terminata nel 1991, in cui si è provveduto alla definizione delle specifiche relative ai servizi base essenziali e ad alcuni servizi supplementari, si è passati ad una seconda fase conclusasi nel 1993 (PHASE 2) durante la quale si sono integrati servizi base e supplementari.

Il servizio venne commercializzato per la prima volta verso la metà del 1991, e nel 1993 esistevano già 36 reti GSM in 22 paesi. Sebbene GSM sia stato standardizzato in Europa, non è uno standard europeo: il sistema GSM (incluso DCS 1800 e PCS1900) è stato progressivamente adottato in oltre 80 paesi sparsi in tutto il mondo.

Fin dall'inizio si volle che il sistema GSM fosse compatibile con il sistema ISDN, sia in termini di servizi offerti che di segnali di controllo adottati. Purtroppo le limitazioni delle trasmissioni radio, sia di banda che di costi, non hanno permesso di raggiungere la velocità di un canale ISDN-B di 64Kbps.

Usando le definizioni ITU-T (*International Telecommunication Union*), i servizi di telecomunicazioni possono essere divisi in: servizi portanti, teleservizi e servizi supplementari.

Il teleservizio fondamentale supportato dal sistema GSM è la comunicazione telefonica. La voce viene codificata in forma digitale e trasmessa attraverso la rete GSM come un flusso di bits. Vi è poi il teleservizio relativo alle chiamate di emergenza (cioè dei numeri di emergenza) che viene considerato un servizio distinto, in quanto l'operatore può consentire ai terminali anche privi di SIM di effettuare chiamate di emergenza.

Gli utenti GSM possono ricevere e trasmettere dati, a velocità finì a 9600 bps, con utenti di reti telefoniche fisse (*Plain Old Telephone Service, POTS*), reti ISDN, reti pubbliche a commutazione di pacchetto, a commutazione di circuito basate su diversi metodi di accesso e protocolli, come X.25 e X.32. essendo GSM una rete digitale, non è richiesto un modem tra l'utente e la rete GSM. È possibile inviare e ricevere fax, come descritto nella raccomandazione T.30 di ITU-T, utilizzando un appropriato adattatore fax.

Una funzionalità particolare di GSM, non offerta dalle vecchie reti analogiche, è il servizio *Short Message Service SMS*. SMS è un servizio bidirezionale che consente di trasmettere e ricevere brevi messaggi (fino a 160 bytes) alfanumerici. I messaggi vengono gestiti dalla rete secondo una strategia store-and-forward. Il servizio SMS può essere usato in modalità punto-punto per mandare un messaggio ad un altro utente GSM, o in modalità *cell-broadcast* per mandare messaggi di aggiornamento sul traffico e notizie.

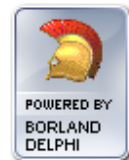
## Modulo di identificazione utente (SIM card)

---

Uno dei componenti più importanti e distintivi del sistema GSM è la cosiddetta SIM, acronimo di *Subscriber Identity Module*, detta anche [SIM card](#). La SIM card è una scheda di memoria estraibile su cui sono memorizzati i dati descrittivi dell'abbonato, compreso il numero di telefono. Trasferendo la SIM card da un telefono all'altro è possibile mantenere tutte le informazioni relative all'abbonamento. Inoltre l'abbonato può anche cambiare operatore, mantenendo lo stesso telefono, semplicemente cambiando SIM card. Alcuni operatori, per contro, inibiscono questa funzionalità, e consentono l'uso di una sola SIM card su ogni terminale (a volte quella emessa da loro stessi): questa pratica, illegale in alcuni paesi, è chiamata *SIM locking*. Negli USA la maggior parte degli operatori, bloccano i terminali da loro venduti. Il motivo risiede nel fatto che il prezzo del terminale è in gran parte sovvenzionato dai profitti dell'abbonamento, e quindi gli operatori cercano di evitare di favorire i concorrenti in caso di migrazione. Gli abbonati hanno il diritto di chiedere di rimuovere il blocco dietro pagamento di una tariffa (cosa che a volte gli operatori fingono di ignorare), o utilizzare altri mezzi *privati* per rimuovere il blocco, come il download da [internet](#) di appositi software. Alcuni operatori USA, come ad esempio la [T-Mobile](#), rimuovono gratuitamente il blocco se l'abbonato ha depositato una cauzione per un certo periodo. Nella maggior parte dei paesi la rimozione del blocco non è considerata illegale.



IL SOFTWARE :



***Il gruppo ha scelto come linguaggio di programmazione : Delphi della casa americana Borland, sotto viene riportato la sua breve storia e le funzioni principali:***

### ***Cos'è Delphi***

Delphi è un ambiente di programmazione visuale ad oggetti per lo sviluppo rapido di applicazioni a carattere generale e di applicazioni client/server per Windows 95 e 98 e Windows NT.

✓ Cronologia dei prodotti Borland

La Borland dopo avere distribuito Turbo Pascal 6.0 per DOS ha introdotto sul mercato un prodotto per Windows 3.1 e lo ha chiamato Borland Pascal 7.0 (versione Windows di Turbo Pascal 7.0). Successivamente ha completamente rivisto la struttura dei suoi compilatori e strumenti di sviluppo accessori introducendo la famiglia Delphi.

A tutt'oggi sono state rilasciate ben 6 edizioni di Delphi :

Delphi 1 per lo sviluppo di applicativi a 16 bit in ambiente Windows 3.1 e 3.11

Nell'ordine si sono poi succedute le versioni di Delphi2 , Delphi3, Delphi4, Delphi5, Delphi6 per lo sviluppo di applicazioni a 32 bit per Windows 9x e Windows NT. I compilatori Borland sono generalmente disponibili in tre versioni :

- a) versione desktop (a basso costo)
- b) versione professional
- c) versione client/server o enterprise

---

*IL LINGUAGGIO PASCAL*

---

Il linguaggio di programmazione di Delphi è basato sul Pascal, introdotto da Niklaus Wirth (università di Ginevra) appositamente come mezzo di insegnamento della programmazione strutturata. Per questo motivo tale linguaggio risulta essere assai elegante e leggibile, soprattutto se lo confrontiamo con linguaggi analoghi come C/C++, più strutturalmente vicino al linguaggio macchina oppure al Basic, che solo in tempi recenti ha subito una profonda riadattamento ai canoni della programmazione strutturata.

La facilità di lettura si accompagna simmetricamente alla facilità di scrittura: Se si tiene conto

del fatto che Delphi è un compilatore velocissimo, scrivere programmi in Pascal per Windows

significa, beneficiare di un ottimo compromesso tra velocità degli eseguibili prodotti e facilità di scrittura del codice. Pascal è un linguaggio procedurale, dotato di tipizzazione forte. Questo sta a significare che per ogni variabile utilizzata all'interno di programmi Pascal deve essere esplicitamente dichiarato un tipo (ad esempio integer, variabile numerica intera; real, variabile numerica reale, ecc.).

# Componenti:

## VaComm



Questo componente permette di mandare comandi alla porta seriale (Com).

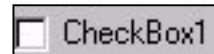
Bisogna specificare due proprietà fondamentali, la speed, che è la velocità con la quale l'oggetto viene interfacciato al computer ; numPort che indica su quale porta 'com' deve mandare i comandi o ricevere comandi.

## File find



Questo componente permette di cercare dei files. Bisogna settare dei parametri, l'estensione dei file che deve cercare, la directory da dove deve partire la ricerca.

## RadioButton e CheckBox



Le classi TRadioButton o, meglio, TRadioGroup, e TCheckBox, sono gli oggetti che rappresentano un gruppo di radio button e di check-box, si prestano per consentire la scelta di opzioni all'interno di un programma. Il gruppo di radio button esprime sempre una serie di scelte in alternativa fra loro. La funzione di risposta attraverso la quale è possibile rilevare l'evento di cambiamento dello stato degli oggetti di interfaccia non è più di tipo OnChange, bensì OnClick. proprietà chiave!

## Button e SpeedButton.

In Delphi i bottoni "classici", quelli per intenderci senza grafica e con un solo stato (non possono, ad esempio, risultare Fondamentalmente l'interazione del programmatore con oggetti appartenenti a questa classe si limita, oltre alla definizione di elementi a design-time come caption, dimensioni e posizione, alla risposta all'evento OnClick. Esistono altre due classi di pulsanti in Delphi, una, chiamata TBitBtn (da bitmap button), che in pratica aggiunge ai semplici bottoni la possibilità di inserire, oltre alla caption (la "scritta") anche un glyph, un elemento grafico in grado di far riconoscere al primo colpo d'occhio le funzionalità associate all'elemento. Le particolarità di questo tipo di pulsanti, comunque, sono principalmente legate all'introduzione della grafica: per quanto riguarda le modalità di utilizzo, risultano perfettamente analoghe a quelle dei "semplici" TButton. Diversamente, la terza classe di pulsanti, TSpeedButton, oltre ad essere personalizzabile con elemento

grafico deciso dall'utente, ha la peculiarità di essere gestita in gruppo per creare toolbar, toolbox e comunque permette di gestire due stati (bottone premuto, bottone alzato), cosa che amplia enormemente gli ambiti di applicazione.

### ***Menu e status-bar.***

Nelle applicazioni professionali spesso si utilizza una status-bar per fornire utili informazioni fondamentali di aiuto all'utente. La costruzione di una status-bar passa attraverso il componente TPanel, un particolare oggetto in grado di occupare un'area rettangolare, eventualmente evidenziata da bordo ed effetti tridimensionali di rilievo o incavo e collocata nella parte inferiore di una Form.



### ***Eventi periodici: la classe TTimer***

Gli oggetti TTimer causano un evento OnTimer con cadenza periodica. La durata del periodo è specificata nella proprietà interval, espressa in millisecondi. All'interno della funzione associata a tale evento è quindi possibile compiere operazioni ripetute ad intervalli di tempo regolari. La proprietà chiave Enabled, di tipo boolean, permette di abilitare o disabilitare temporaneamente il lancio degli eventi OnTimer.

## ***Conclusioni***

*Delphi permette di creare degli applicati molto velocemente grazie anche ad una palette di componente molto assortita.*

## *Spiegazione Generale del software :*

Il software da noi creato permette ad una persona diversamente abile, nel nostro caso Fabrizio, di leggere SMS e anche di inviarne, tramite l'utilizzo di un unico pulsante, superando tutte le difficoltà per una persona diversamente abile di usare un cellulare.

### Lettura SMS

---

Il software di lettura è basato su due programmi.

Il primo scarica dal modem, nel quale è inserita una sim card, gli SMS li decodifica e poi li salva in una cartella. A questo punto parte il programma di sintesi vocale che va a vedere se nella cartella nella quale vengono salvati gli SMS ce ne sono di nuovi, se ne trova, il programma di sintesi vocale ( ' ER Finestra ' ) incomincia a leggerli.

### Invio

---

Il software di invio praticamente richiama un file di testo di nome 'Msg.txt' il quale nella prima linea contiene il numero di telefono del destinatario al quale verrà inviato il messaggio, mentre dalla seconda linea in poi il testo del messaggio (max 160 caratteri).

# Programma di Lettura

Il software di lettura è basato su due programmi :

- pMachStatus.exe
- SMS\_speak.exe

Serve per scaricare i messaggi dal modem GSM. Questo programma è stato fatto incorporando una macchina a stati finiti.

Per creare questo applicativo abbiamo utilizzato i seguenti oggetti :

- 2 Memo
- 1 TRadioGroup
- 1 Timer
- 1 Log
- 1 VaComm

### Da Wikipedia, l'enciclopedia libera.

Una FSM (Finite State Machine/Macchina a Stati Finiti) è un dispositivo, o un suo modello in forma di macchina sequenziale, creato per eseguire un particolare compito, che può trovarsi in diverse **configurazioni** più o meno complesse caratterizzate primariamente da una variabile che appartiene ad un determinato insieme di **stati** , e che evolve in base agli stimoli od ordini ricevuti in ingresso schematizzati da simboli appartenenti ad un determinato alfabeto.

Quando la macchina a stati si trova in un dato stato, essa può accettare solo un sottoinsieme dei simboli del suo alfabeto. L'evoluzione del suo funzionamento parte da un particolare stato detto **stato iniziale**. Un sottoinsieme privilegiato dei suoi stati è detto insieme degli **stati finali** o *marcati*.

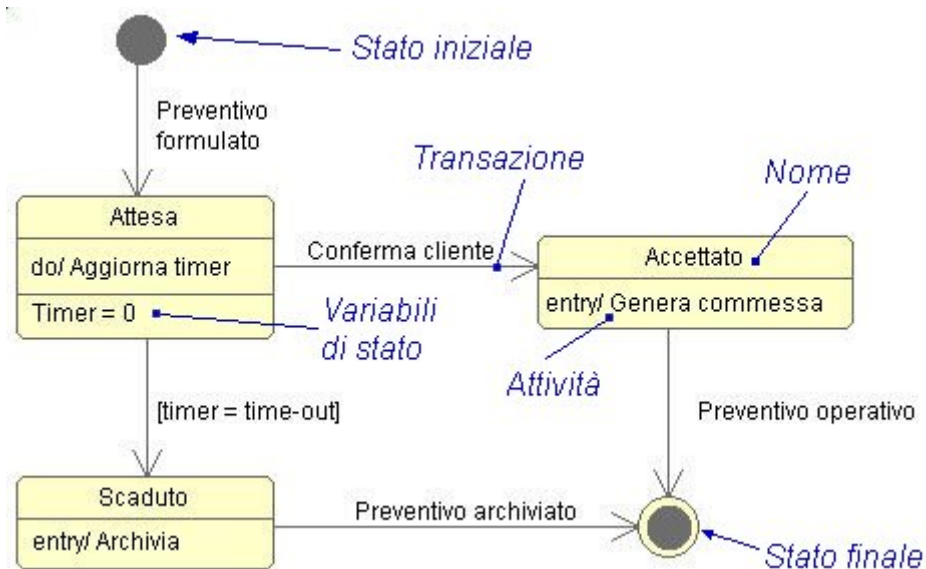
Si definisce anche come un sistema dinamico (Teoria dei sistemi) (se evolve nel tempo), discreto (nella scansione del tempo e nella descrizione del suo stato) e invariante (il sistema si comporta alla stessa maniera indipendentemente dall'istante di tempo agisce).

In genere gli automi sono **deterministici**, ovvero dato uno stato ed un simbolo in ingresso è possibile una sola transizione.

Il funzionamento di una FSM può essere descritto attraverso una rappresentazione grafica denominata grafo o grafico.

In figura 1 è riportato il diagramma degli stati del ciclo di vita di un preventivo. Un preventivo non appena redatto si trova in uno stato di "attesa responso". A fronte di una

conferma da parte del cliente transita nello stato di "accettato", mentre, nel caso in cui ciò non avvenga e trascorso un lasso di tempo maggiore del periodo di validità concordato, il preventivo transita nello stato di "rifiutato".



**Figura 1** State diagram del ciclo di vita di un preventivo.

Dall'esempio, si può notare che il diagramma degli stati coinvolge due prospettive di dinamicità, per così dire, esterna ed interna. La prima è una vera e propria interazione, ossia viene descritto il comportamento esterno dell'oggetto e la sua interazione con la restante parte del sistema, realizzata per mezzo di scambio esplicito di messaggi. Per esempio, si consideri ancora il diagramma di figura 1, quando un preventivo transita nello stato accettato, automaticamente si esegue la funzione "genera una commessa", ossia viene inviato un messaggio ad un opportuno oggetto che provvederà ad avviare un nuovo flow.

La dinamicità "interna" è invece rappresentata dal cambiamento di stato, descritto per esempio, dall'attribuzione di nuovi valori agli attributi dell'oggetto.

## *Macchina a Stati Finiti*

- STATO 0 :** Stato di partenza.  
Esegue il controllo sul centro servizi memorizzato all'interno del modem GSM, che deve essere uguale a quello presente nel file Setting\_sms.ini. Nel caso in cui i due parametri siano diversi, il pocket memorizza il numero di centro servizi che ricava dal nome dell'operatore presente nel file Setting\_sms.ini. Questa operazione serve nel caso si cambi la sim card, passando ad un altro operatore. Pertanto bisogna scrivere all'interno del file il nome del nuovo operatore.
- STATO 1:** Come prima cosa esegue un controllo sulla variabile (di tipo integer) C se è uguale a 1 passa direttamente allo stato 8, mentre se la C è uguale a 0 esegue la seguente operazione.  
Invia al modem GMS il comando: AT+CMGL="ALL" , che serve per controllare se ci sono dei nuovi messaggi.
- STATO 2:** Attendo la risposta dal modem dopo avergli inviato il comando, se non ricevo una risposta il led comincia a lampeggiare, altrimenti passo allo stato successivo.
- STATO 3 :** Controllo la risposta del modem, se all'interno trovo +CMGL: incremento la variabile C e passo allo stato 1, questo vuol dire che non ci sono nuovi messaggi; altrimenti pongo la variabile C a 0 e passo al prossimo stato.
- STATO 4 :** A questo punto salvo la risposta del modem in un file chiamato '[Temp0.txt](#)' .  
Il file che ho appena salvato contiene il messaggio di testo, ma anche altre informazioni che non sono importanti per cui modifico il file '[Temp0.txt](#)' e ne creo un altro pulito '[Temp2.txt](#)'.  
Adesso decodifico il messaggio e lo salvo all' interno della cartella 'Sms'  
Esempio: [2005 06 09 14 15 58](#)
- STATO 5 :** Adesso cancello i messaggi mandando il comando AT+CMGD=0,4 che elimina all'interno della sim tutti i messaggi.
- STATO 6 :** Attendo risposta dal Modem che la cancellazione sia finita;  
Quando l'ho ricevuta passo allo stato 1 per controllare che non ci siano più messaggi;
- STATO 7 :** Chiusura applicazione.
- STATO 8 :** Chiudo questo programma e apro il programma di sintesi vocale.

Quando il programma Delphi inizia la sua esecuzione, si verificano nell'ordine de eventi: l'evento OnCreate e quello OnShow

## **OnCreate**

In questa fase si inizializza il programma. Come prima cosa si portano le variabili C, Status (di tipo integer) a 0, poi assegno alla proprietà RG.ItemIndex del TRadioGroup lo stato della variabile Status. Cancello la memo1, e assegno alla variabile pathsms (di tipo string) il percorso dove deve salvare gli sms.

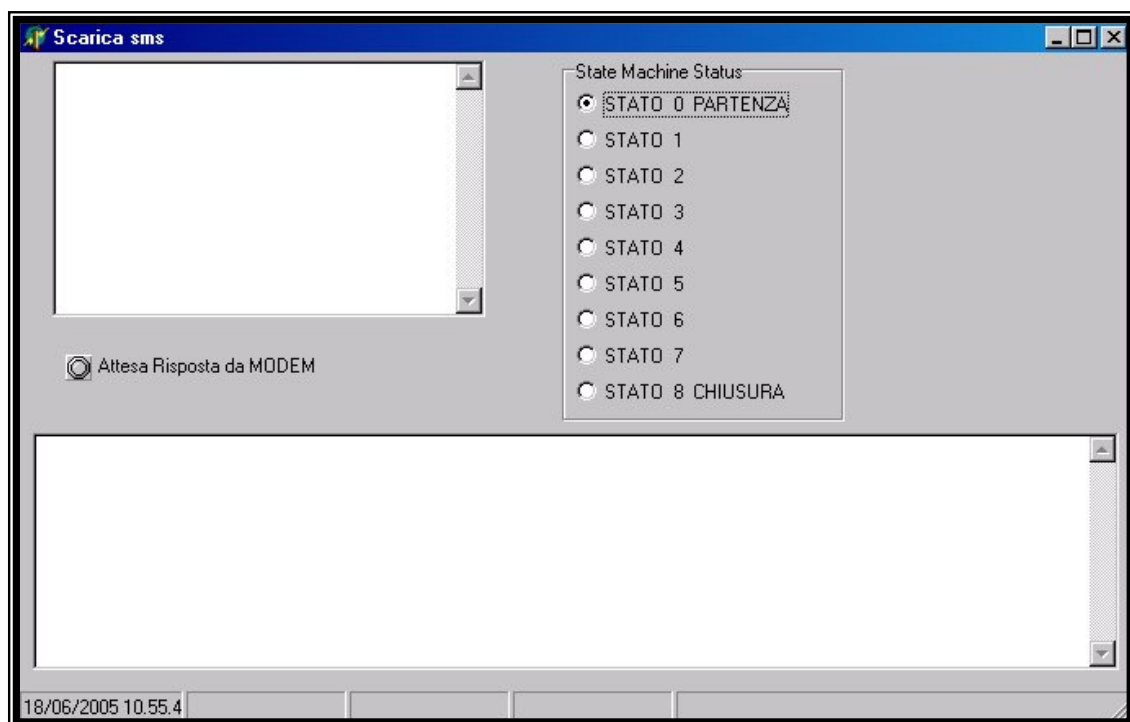
## OnShow

Qui controllo vari parametri, appoggiandomi sul file Setting\_sms.ini.

Setto la proprietà PortNum, dell'oggetto VaComm, questo permette al programma di sapere su quale porta com deve interfacciarsi per comunicare con il modem. Poi setto la proprietà Baudrate sempre dell'oggetto VaComm che sarebbe la velocità con la quale il computer comunica con il modem. Assegno alla variabile Centro\_serv\_ini (di tipo string) il nome dell'operatore, che servirà poi nello stato 0. Controllo inoltre che nel computer sia presente la porta Com tramite la function IsComPortInstalled, nel caso in cui non ci fosse la porta viene emesso un messaggio di errore, mentre se la porta è presente, apre la comunicazione (VaComm1.open).

In base al nome contenuto nella variabile Centro\_serv\_ini, assegno alla variabile Centro\_serv\_ini\_n (di tipo integer) il numero del centro servizi.

## Il programma in esecuzione si presenta così



**unit uMachStatus;**

**interface**

**uses**

**Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
ExtCtrls, StdCtrls, FileFind, IniFiles, ALed, VaClasses, VaComm,  
ComCtrls, LogFile, MMSystem, shellapi;**

**type**

**TForm1 = class(TForm)**

**rg: TRadioGroup;**

**TimFSM: TTimer;**

**Memo1: TMemo;**

**led1: ThhALed;**

**Label2: TLabel;**

**sbar: TStatusBar;**

**VaComm1: TVaComm;**

**Memo2: TMemo;**

**Log: TLogFile;**

**procedure FormCreate(Sender: TObject);**

**procedure TimFSMTimer(Sender: TObject);**

**procedure FormShow(Sender: TObject);**

**procedure FormClose(Sender: TObject; var Action: TCloseAction);**

**procedure VaComm1RxChar(Sender: TObject; Count: Integer);**

**private**

**{ Private declarations }**

**public**

**{ Public declarations }**

**STATUS, c, RMOD : Integer;**

**Function IsComPortInstalled( port : integer): Boolean;**

**procedure Richiesta\_SMS;**

**procedure Decodifica\_SMS;**

**function Rubrica(n: string): string;**

**procedure Separa\_mex;**

**procedure Cancella\_SMS(n:integer);**

**end;**

**var**

**Form1: TForm1;**

**Incod, r, fname, dest, pathsms: string; // luogo di nascita ; codice città  
codice: word;**

**k, b,c,m: integer;**

**nc: int64;**

**z: Tinifile;**

**bufferpieno: boolean;**

**tl: Tstringlist;**

**nmex, nmext, nmext2: integer;**

```
s2,Centro_serv_ini,Centro_serv_pocket,Comando : string;
```

implementation

```
{ $R *.DFM }
```

```
//
```

```
Function TForm1.IsComPortInstalled( port : integer): Boolean;
```

```
var
```

```
  MaxPorts    : integer;
```

```
  hPort       : THandle;
```

```
  PortNumber  : integer;
```

```
  PortName, st : string;
```

```
Begin
```

```
  st := '';
```

```
  { Verifica della piattaforma }
```

```
  case Win32Platform of
```

```
    VER_PLATFORM_WIN32_NT: MaxPorts := 256;
```

```
    VER_PLATFORM_WIN32_WINDOWS: MaxPorts := 9;
```

```
  end;
```

```
  for PortNumber := 1 to MaxPorts do
```

```
  Begin
```

```
    if PortNumber > 9 then
```

```
      PortName := '\\.\COM' + IntToStr( PortNumber ) // ask Microsoft why...
```

```
    else
```

```
      PortName := 'COM' + IntToStr( PortNumber );
```

```
    hPort := CreateFile(PChar( PortName ),
```

```
      GENERIC_READ or GENERIC_WRITE,
```

```
      0, nil, OPEN_EXISTING, 0, 0 );
```

```
    // note that ports already in use by other apps
```

```
    // will *NOT* be detected here
```

```
    if not ( hPort = INVALID_HANDLE_VALUE )
```

```
      then st := st +PortName+'';
```

```
    CloseHandle( hPort );
```

```
  end;
```

```
  Result := Pos('COM'+IntToStr(port),st) <> 0;
```

```
end;
```

```
//
```

```
function strFileLoad(const aFile: string): string;
```

```
var
```

```
  aStr: TStringList;
```

```
Begin
```

```
  Result := '';
```

```
  aStr := TStringList.Create;
```

```
  try
```

```

    aStr.LoadFromFile(aFile);
    Result := aStr.Text;
finally
    aStr.Free;
end;
end;

//
// Lettura degli SMS
//
procedure TForm1.Richiesta_SMS;
var s0: string;
Begin
    memo2.Clear;
    sbar.Panels[2].Text := 'COMANDO';
    s0 := 'AT+CMGL="ALL"' + Chr(13);
    VaComm1.WriteText(s0);
end;

Function mese(n: integer): string;
Begin
    case n of
        1: Result := 'Gennaio';
        2: Result := 'Febbraio';
        3: Result := 'Marzo';
        4: Result := 'Aprile';
        5: Result := 'Maggio';
        6: Result := 'Giugno';
        7: Result := 'Luglio';
        8: Result := 'Agosto';
        9: Result := 'Settembre';
        10: Result := 'Ottobre';
        11: Result := 'Novembre';
        12: Result := 'Dicembre';
    end;
end;

// Ricerca all'interno della rubrica
function TForm1.Rubrica(n: string): string;
var a:Tstringlist;
    k,d:integer;
    st:string;
Begin
    a:=Tstringlist.Create;
    a.LoadFromFile('rubrica.txt');
    d:=0;
    for k:=1 to a.count-1 do
        Begin
            if (pos(n,a.strings[k])<>0)
                then d:=k;
        end;
    end;
end;

```

```

if d=0
  then
    Begin
      st:='';
      for k:=1 to length(n) do
        st:=st+n[k]+'.';
      end
    else
      Begin
        st:=a.strings[d];
        d:=pos('=',st);
        delete(st,1,d);
      end;
    result:=st;
    a.Free;
end;

//

procedure TForm1.Cancella_SMS(n:integer);
var
  s1 : string;
Begin
  s1 := 'AT+CMGD='+IntToStr(n)+chr(13);
  VaComm1.WriteText(s1);
  //memo1.Lines.Add('Messaggi cancellati');
end;

procedure TForm1.Separa_mex;
var
  nmem, k, n: integer;
  st : string;
  tl2,tl : TStringlist;
Begin
  tl := TStringList.Create;
  Tl2 := TStringList.Create;
  memo2.Lines.SaveToFile('temp0.txt');
  //pulizia testa
  k := 0;
  st := memo2.Lines.Strings[k];
  while Pos('+CMGL:', st) = 0 do
    Begin
      memo2.Lines.Delete(k);
      //inc(k);
      st := memo2.Lines.Strings[k];
    end;
  memo2.Lines.SaveToFile('temp1.txt'); // cancellare
  //pulizia coda
  k := memo2.Lines.Count -1;
  st := memo2.Lines.Strings[k];
  while Pos('OK', st) = 0 do

```

```

Begin
  memo2.Lines.Delete(k);
  dec(k);
  st := memo2.Lines.Strings[k];
  end;
memo2.Lines.Delete(k);
//finisce pulizia coda
memo2.Lines.SaveToFile('temp2.txt');
tl2.LoadFromFile('temp2.txt');
k := 0;
nmex := 1;
n := tl2.Count;
repeat
  st := tl2.Strings[k];
  if k = 0
  then tl.Add(st)
  else
  Begin
    if Pos('+CMGL:', st) <> 0
    then
    Begin
      tl.SaveToFile('mex'+IntToStr(nmex)+'.txt');
      tl.Clear;
      inc(nmex);
    end;
    tl.Add(st);
  end;
  inc(k);
until (k > n-1);
tl.SaveToFile('mex'+IntToStr(nmex)+'.txt');
tl.Free;
tl2.Free;
nmext := nmex;
nmext2 := nmex;
decodifica_sms;
//inizio a cancellare i mex provvisori
repeat
  deletefile('mex'+IntToStr(nmex)+'.txt');
  dec(nmex);
until nmex < 1;
//Finish
end;

```

```

procedure TForm1.Decodifica_SMS;
var
  nmem, k, n: integer;
  tipo, mittente, Data, ora,
  data_ora, testo, st, data1,
  ora1, data2, ora2, me: string;
  tl2,tl : Tstringlist;

```

## Begin

```
tl := Tstringlist.Create;
tl2 := Tstringlist.Create;
repeat
  tl2.LoadFromFile('mex'+IntToStr(nmext)+'.txt');
  tl.LoadFromFile('mex'+IntToStr(nmext)+'.txt');
  k := 0;
st := tl2.Strings[k];
Delete(st, 1,6);
n := Pos(',', st);
nmem := StrToInt(Copy(st, 1,n - 1));
// si estrae il tipo di messaggio.
Delete(st, 1,n);
n := Pos(',', st);
tipo := Copy(st, 2,n - 3);
Delete(st, 1,n);
// si estrae il num. di telefono
n := Pos(',', st);
mittente := Copy(st, 2,n - 3);
Delete(st, 1,n);
// si estrae la data
n := Pos(',', st);
Data := Copy(st,2,8);
data1 := '20' + Copy(Data, 1,2) + '_' + Copy(Data, 4,2) + '_' + Copy(Data, 7,2);
//Serve per modificara la data in anno_mese_giorno
me := Mese(StrToInt(Copy(Data,4,2)));
if copy(data, 7,1) = '0'
  then data2 := Copy(Data, 8,1) + ':' + Me + ':20' + Copy(Data, 1,2)
  else data2 := Copy(Data, 7,2) + ':' + Me + ':20' + Copy(Data, 1,2);
//Serve per modificara la data per l'erfinestra
Delete(st, 1,10);
// si estrai l'ora
ora := Copy(st, 1,8);
ora1 := Copy(ora, 1,2) + '_' + Copy(ora, 4,2)+'_' + Copy(ora, 7,2) ;
//Serve per modificare l'ora in ore_minuti_secondi
ora2 := Copy(ora, 1,2) + 'e' + Copy(ora, 4,2);
//Serve per modificare l'ora per l'erfinestra
data_ora := data1 + '_' + ora1;
n := tl2.Count;
tl2.Clear;
tl2.Add('Inizio lettura messaggio');
tl2.Add('MITTENTE: ' + rubrica(mittente));
tl2.Add('RICEVUTO IL GIORNO:' + data2 + ' alle ore:' + ora2);
tl2.Add(':Testo messaggio');
inc(k);
repeat
  st := tl.Strings[k];
  tl2.Add(st);
  inc(k);
until (k > n-1);
tl2.Add(':Fine messaggio');
```

```

tl2.SaveToFile(pathsms + data_ora + '.txt');
dec(nmext);
until nmext < 1;
tl.Free;
tl2.Free;
end;

// *****
procedure TForm1.FormCreate(Sender: TObject);
Begin
  Status := 0;
  c := 0;
  RMOD := 0; // Risposta Modem
  RG.ItemIndex := Status;

  // segue old
  // Form1.Caption := 'SMS Programma base per la gestione degli SMS';

  k := 0;
  b := 0;
  Memo1.Clear;

  pathsms:= ExtractFilePath(Application.ExeName)+'sms\';

  sbar.Panels[0].Text := DateTimeToStr(Now);
  bufferpieno := False; //da togliere

end;

//
// Gestione Cambio Stati FSM
//
procedure TForm1.TimFSMTimer(Sender: TObject);
  var n : integer;
Begin
  Case STATUS of
    0 : Begin
      if m = 1
      then
        Begin
          if Pos(Centro_serv_ini,memo2.Text) <> 0
          then m:= 0//Showmessage('Centro servizi uguale')
          else
            Begin
              //Showmessage('Centro servizi non uguale');
              Comando := '';
              Comando := 'AT+CSCA="'+ Centro_serv_ini+Chr(13);
              VaComm1.WriteText(Comando);
              Sleep(2000);
              Comando := '';
              Comando := 'AT+CMGF=1'+Chr(13);
            End;
          End;
        End;
      End;
  End;

```

```

    VaComm1.WriteText(Comando);
    Sleep(2000);
    Comando := '';
    Comando := 'AT$SCRT' + Chr(13);
    VaComm1.WriteText(Comando);
    m := 0;
    End;
end;
Inc(Status);
End;

```

```

1 : Begin
  If c = 1
  Then
  Begin
    Status := 8; // Chiudo dopo due tentativi
  End
  Else
  Begin
    Richiesta_SMS;
    Memo1.Lines.Add('Richiesta messaggi al modem ');
    // Button1.Enabled := True;
    Inc(Status);
  End;
  End;
2 : Begin
  if pos('OK',memo2.text) <> 0
  then inc(rmod);
  If RMOD <> 0
  Then
  Begin
    Memo1.Lines.Add('MODEM: richiesta messaggi ricevuta');
    Led1.Value := False;
    Inc(Status);
    //RMOD := 0;
    //Button1.Enabled := False;
  End
  Else
  Begin
    Led1.Value := True;
  end;
  End;
3 : Begin
  if pos('+CMGL:',memo2.text) = 0
  Then
  Begin
    Memo1.Lines.Add('Nessun SMS');
    Inc(c);
    Status := 1;
  End

```

```

Else
  Begin
    Memo1.Lines.Add('Presenti SMS nel MODEM');
    c := 0;
    Inc(Status);
  End;
End;

4 : Begin
  Memo1.Lines.Add('Interpretazione Messaggi ');
  Separa_mex;
  Inc(Status);
End;
5 : Begin
  rmod := 0;
  memo2.Clear;
  Memo1.Lines.Add('INVIO Cancellazione messaggi dal modem ');
  n := nmext2;
  Cancella_SMS(n);
  dec(nmext2);
  //Button1.Enabled := True;
  Inc(Status);
  //c:=2;
End;
6 : Begin
  if nmext2 > 0
    then Status := 5
  else
    Begin
      if pos('+CMGD=',memo2.text) <> 0
        then inc(rmod);
      If RMOD <> 0
        Then
          Begin
            Memo1.Lines.Add('MODEM: cancellazione messaggi avvenuta');
            Led1.Value := False;
            Status := 1;
            RMOD := 0;
            // Button1.Enabled := False;
          End
        Else Led1.Value := True;
      end;
    End;
7 : Begin
  Memo1.Lines.Add('CHIUSURA APPLICAZIONE');
End;
8 : Begin
  Close;
end;
End;
rg.ItemIndex := Status;

```

```

end;

procedure TForm1.FormShow(Sender: TObject);
  var fname : string;
      speed,Centro_serv_ini_n,n: integer;
Begin
  //application.Minimize;

  PlaySound('ContrMsgIniz.wav', 0, SND_ASYNC OR SND_FILENAME);

  m := 0 ;
  Centro_serv_ini_n := 0;
  tl:=Tstringlist.Create;
  if Pos('Q',LowerCase(ParamStr(1))) <> 0 Then Close;

  fname := ExtractFilePath(Application.ExeName);
  n := length(fname);
  fname := copy(fname,1,n-8) + 'Setting_sms.ini';

  Z := TiniFile.Create(fname);
  VaComm1.PortNum := Z.ReadInteger('MODEM', 'COMPORT',0);
  Z.WriteString('MESSAGGI', 'LOPEN', DateTimeToStr(Now));
  Centro_serv_ini := Z.ReadString('MODEM','CENTRO_SERVIZI','');
  speed := Z.ReadInteger('MODEM','SPEED',9600);
  case speed of
    9600 : VaComm1.Baudrate := br9600;
    19200 : VaComm1.Baudrate := br19200;
  end;
  Z.Free;

  // Verificare Presenza della porta Seriale
  If NOT IsComPortInstalled(VaComm1.PortNum)
  Then
    Begin
      log.LogToFile(DateTimeToStr(Now)+' COM'+IntToStr(VaComm1.PortNum)+' HDW
ERROR');
      Close;
    End
  else VaComm1.open;

  If Centro_serv_ini = 'VODAFONE'
  then Centro_serv_ini_n := 1;
  If Centro_serv_ini = 'TIM'
  then Centro_serv_ini_n := 2;
  If Centro_serv_ini = 'WIND'
  then Centro_serv_ini_n := 3;

  Case Centro_serv_ini_n of
    1 : Centro_serv_ini := '+393492000200''; // Omnitel
    2 : Centro_serv_ini := '+393359609600''; // Tim

```

```
3 : Centro_serv_ini := '+393205959100''; // Wind  
End;
```

```
Comando := 'AT+CSCA?'+Chr(13);  
VaComm1.WriteText(Comando);
```

```
end;
```

```
procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);  
Begin  
  Tl.Free;  
  log.LogToFile(TimeToStr(Now)+' CHIUSURA APPLICAZIONE');  
  fname := ExtractFilePath(application.ExeName) + 'SMS_speak.exe';  
  ShellExecute(handle,pchar('open'),Pchar(fname),nil,nil,SW_SHOWNORMAL);  
  sleep(2000);  
end;
```

```
procedure TForm1.VaComm1RxChar(Sender: TObject; Count: Integer);  
Begin  
  r := r + VaComm1.ReadText;  
  if (Pos('OK', r) <> 0) or  
     (Pos('ERROR', r) <> 0) or  
     (Pos('>', r) <> 0) then  
    Begin  
      memo2.Lines.Add(R);  
      Inc(m);  
      r := '';  
      sbar.Panels[2].Text := 'RX CONCLUSA';  
      bufferpieno := True;  
    end;
```

```
end;
```

```
end.
```

## *SINTESI VOCALE DEGLI SMS*

---

Il software SMS\_speak.exe viene avviato quando il programma pMachStatus.exe si chiude.

Il programma utilizza, per leggere i messaggi precedentemente salvati, il sintetizzatore vocale 'Er Finestra' scaricabile liberamente da internet. Dopo che SMS\_Speak.exe è stato avviato, viene effettuato un controllo di quanti messaggi sono presenti all'interno della cartella 'Sms', distinguendo fra messaggi già letti e quelli ancora da leggere. In presenza di nuovi messaggi il sintetizzatore vocale provvede alla loro lettura.

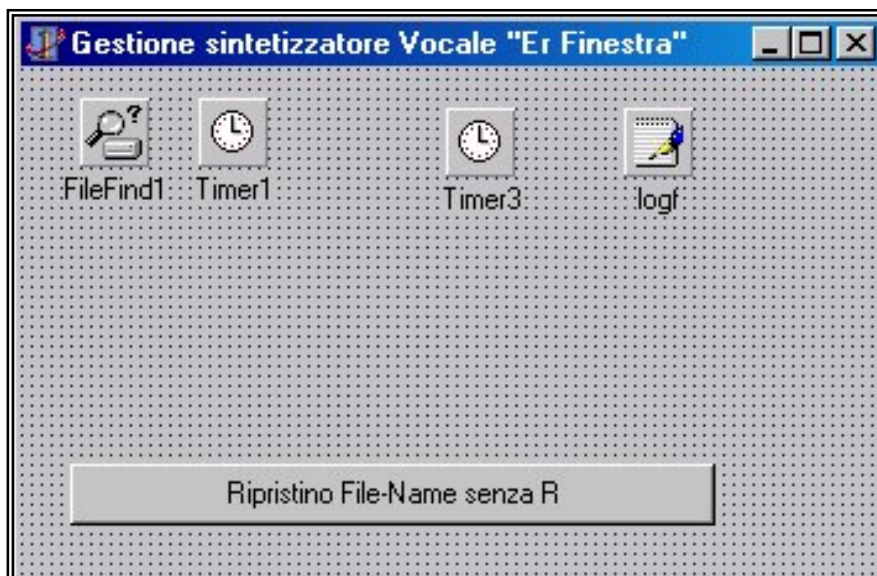
Per creare questo applicativo abbiamo utilizzato i seguenti oggetti :

- 2 Timer
- 1 Log
- 1 FileFind
- 1 Button

Il software viene avviato solo se il programma Master si trova nello stato 'Stand by'.

Come prima cosa il programma controlla che l' 'Er Finestra' non sia già in esecuzione, se così non fosse avvia l' 'Er Finestra'. Per avviare il programma di sintesi vocale si usano funzioni di tipo API (Application Program Interface), che si basano sull' handle (numero intero associato da windows ad una qualsiasi finestra presente) del programma. Cerca all'interno della cartella 'Sms' quanti file sono presenti, dice quanti file ha trovato già letti, e quanti deve ancora leggerne. Se trova dei nuovi file da leggere trasferisce tutto il testo del file all'interno del programma di sintesi vocale, usando ancora funzioni di tipo API. Quando un file viene letto, al suo nome viene aggiunta in testa la lettera R, in modo tale che alla prossima ricerca il software lo consideri già letto. Se il programma Master passa allo stato 'Running', viene colusa la sintesi vocale in corso e il programma viene chiuso.

### **Il programma si presenta così:**



**unit** uSMS\_speak;

**interface**

**uses**

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, ExtCtrls, FileFind, IniFiles, LogFile, MMSystem;

**type**

TForm1 = class(TForm)

FileFind1: TFileFind;

Timer1: TTimer;

Button2: TButton;

Timer3: TTimer;

logf: TLogFile;

procedure FormCreate(Sender: TObject);

procedure FormClose(Sender: TObject; var Action: TCloseAction);

procedure FormShow(Sender: TObject);

procedure FileFind1Terminate(Sender: TObject);

procedure Timer1Timer(Sender: TObject);

procedure Button2Click(Sender: TObject);

procedure Timer3Timer(Sender: TObject);

**private**

{ Private declarations }

procedure SetEditText(ClassName, WindowName, Text : string; EditNr: Integer);

**public**

{ Public declarations }

Function Click\_Bottone(ClassP, WinP, BText, BClass : string): Boolean;

Function Stat\_Bottone(ClassP, WinP, BText, BClass : string): integer;

Procedure LeggiSMS;

procedure TrasferimentoMessaggi;

Function TrovoErFinestra: string;

procedure Comando\_Leggi;

procedure Comando\_Esci;

procedure Comando\_Pausa;

procedure Comando\_Stop;

procedure EliminaR;

function Stato\_master : boolean;

**end;**

**var**

Form1: TForm1;

BH : THandle;

BTnText, BtnClass, pathsms, fname, param : string;

k, w, r, m : integer;

TL1, TL2 : TStringList;

**implementation**

{ \$R \*.DFM }

//

```

// Ricerca del sintetizzatore Vocale sul PC
// dove è installato il programma
//
Function TForm1.TrovoErFinestra : string;
  var z : TiniFile;
      cmd : string;
Begin
  Z := TiniFile.Create(fname);
  cmd := Z.ReadString('ERFINESTRA','PATH','');

  If NOT FileExists(cmd) Then
    Begin
      FileFind1.FilePattern := 'ER FINESTRA.EXE';
      FileFind1.Recursive := True;
      FileFind1.StartDir := 'C:\';
      FileFind1.Execute; //Use the Filesearch
      CMD := FileFind1.FilesFound[0];
      Z.WriteString('ERFINESTRA','PATH',cmd);
      logf.LogToFile(TimeToStr(now)+' AGGIORNO INI PATH= '+CMD);
    End;
    Z.Free;
    Result := cmd;
  End;

procedure TForm1.FormCreate(Sender: TObject);
var
  hparent : HWND;
  cmd : string;
begin
  if not Stato_master
    then application.Terminate;
  logf.LogFile := ExtractFilePath(Application.ExeName) + 'sms_speak.log';

  logf.LogToFile(TimeToStr(now)+' APRO SMS-SPEAK');

  pathsms:= ExtractFilePath(Application.ExeName)+'sms\';
  fname := ExtractFilePath(Application.ExeName) + 'messaggi.ini';

  cmd := TrovoErFinestra;

  // trovo handle della finestra padre
  hparent := FindWindow(PChar('#32770'), PChar('Er Finestra'));
  // Devo evitare che all'avvio dell'appl. Delphi, più copie del
  // pgm Er Finestra siano caricate in memoria
  if (hparent = 0) and (Stato_master)
    Then // Avvio minimizzato l'eseguibile Er Finestra
      WinExec(PChar(cmd), SW_MINIMIZE);

  TL1:= TStringList.Create;
  TL2:= TStringList.Create;
  TL1.Clear;
end;

procedure TForm1.FormClose(Sender: TObject; var Action: TCloseAction);
begin

```

```

logf.LogToFile(TimeToStr(now)+' CHIUDO SMS-SPEAK');

// Click su bottone ESCI
Try
  Comando_Esci ;
except
  logf.LogToFile(TimeToStr(now)+' ERRORE DI CHIUSURA');
end;
T11.Free;
TL2.Free;
end;

procedure TForm1.FormShow(Sender: TObject);
var
  h: HWND;
begin
  application.Minimize;
  param := UpperCase(ParamStr(1));
  If Pos('R',param) <> 0
    Then EliminaR;

  h := FindWindow('#32770', PChar('Er Finestra'));
  If h = 0
    Then
      Begin
        // ER FINESTRA NON TROVATO
        logf.LogToFile(TimeToStr(now)+' ER FINESTRA NON TROVATO');
        Close;
      End
    Else
      Begin
        // Ricerca di tutti i files *.txt nella catella "SMS\"
        filefind1.StartDir := ExtractFilePath(Application.ExeName)+'SMS';
        filefind1.FilePattern:='*.txt'; //ricerca files che hanno l'estensione *.txt
        filefind1.ThreadExecute;
      End;
end;

// II) GetWindow

{
  FindWindowByTitle returns the handle of a window that
  contains a certain "WindowTitle".
}

function FindWindowByTitle(WindowTitle: string): Hwnd;
var
  NextHandle: Hwnd;
  NextTitle: array[0..260] of char;
begin
  // Get the first window
  NextHandle := GetWindow(Application.Handle, GW_HWNDFIRST);
  while NextHandle > 0 do
    begin
      // retrieve its text
      GetWindowText(NextHandle, NextTitle, 255);
    end;
end;

```

```

if Pos(WindowTitle, StrPas(NextTitle)) <> 0 then
begin
    Result := NextHandle;
    Exit;
end
else
    // Get the next window
    NextHandle := GetWindow(NextHandle, GW_HWNDNEXT);
end;
Result := 0;
end;

{
    Example to search for Edit field nr. x
    in another application and send a text to it
}

function FindControlByNumber(hApp: HWND;
    ControlClassName: string;
    ControlNr: Word = 1): HWND;

var
    i: Word;
    hControl: HWND;
begin
    Result := 0;
    if IsWindow(hApp) then
        begin
            Dec(ControlNr);
            hControl := 0;
            for i := 0 to ControlNr do
                begin
                    hControl := FindWindowEx(hApp, hControl, PChar(ControlClassName), nil);
                    if hControl = 0 then Exit;
                end;
            end;
            Result := hControl;
        end;
end;

// procedure SetEditText(ClassName,WindowName,Text : string; EditNr: Integer);
// ClassName,WindowName Nome della classe e della finestra padre
// Text : testo da inviare
// EditNr: numero del controllo Edit su cui operare

procedure TForm1.SetEditText(ClassName,WindowName,Text : string; EditNr: Integer);
var
    hparent, hEdit: HWND;
begin
    // trovo handle della finestra padre
    hparent := FindWindow(PChar(ClassName), PChar(WindowName));
    if hparent <> 0
        then
            Begin
                // trovo handle della finestra figlia 'Edit' nr.: 1
                hEdit := FindControlByNumber(hparent,'Edit', EditNr);
                if hEdit <> 0 then
                    // Test: Send Text to the Edit Field

```

```

    SendMessage(hEdit, WM_SETTEXT, 0, Integer(PChar(Text)));
End

else
  Begin
    logf.LogToFile(TimeToStr(now)+' ERRORE SetEditText');
    logf.LogToFile('Classe: '+ClassName+' Finestra: '+WindowName+' NON TROVATA');
  End;
end;

```

```

procedure TForm1.TrasferimentoMessaggi;
begin
  // Trasferisco il Testo dalla TL1 alla Finestra Applicativo
  SetEditText('#32770','Er Finestra',TL1.Text,1);

  // Eseguo la interpretazione vocale
  Comando_Leggi;
end;

```

(\*  
**The FindWindow function retrieves the handle to the top-level window whose class name and window name match the specified strings. This function does not search child windows.**

```

HWND FindWindow(
    LPCTSTR lpClassName, // pointer to class name
    LPCTSTR lpWindowName // pointer to window name
);

```

#### Parameters

##### lpClassName

**Points to a null-terminated string that specifies the class name or is an atom that identifies the class-name string. If this parameter is an atom, it must be a global atom created by a previous call to the GlobalAddAtom function. The atom, a 16-bit value, must be placed in the low-order word of lpClassName; the high-order word must be zero.**

##### lpWindowName

**Points to a null-terminated string that specifies the window name (the window's title). If this parameter is NULL, all window names match.**

#### Return Values

**If the function succeeds, the return value is the handle to the window that has the specified class name and window name.**

**If the function fails, the return value is NULL. To get extended error information, call GetLastError.**

\*)

```

// funzione di Callback

```

```

(*)
//-----

/* Questa funzione utente di CALLBACK viene chiamata ciclicamente
da Windows a seguito di richiesta da "EnumChildWindow". Ad ogni
chiamata la funzione riceve in successione da Windows l'handler
di un elemento della finestra padre precedentemente trovata.
*/
BOOL CALLBACK Processa_Controlli_Figlio(HWND hwnd, LPARAM lParam)
{ int flg;
  RECT R_Tmp;
  static char buffer[256];

  *buffer = 0;

  // CON "SendMessage WM_GETTEXT" froze problem: quindi uso "GetWindowText"
  // SendMessage(hwnd, WM_GETTEXT, sizeof(buffer)-2, (long)buffer);

  GetWindowText(hwnd, buffer, sizeof(buffer)-2);

  if (*buffer)
    flg = !strcmp(buffer, Titolo_Controllo_Figlio_Cercato);
  else flg = 0;

  if (flg) // Label controllo OK ?
  {
    // Sì: ora verifichiamo la classe

    *buffer = 0;

    GetClassName(hwnd, buffer, sizeof(buffer)-2);

    if (!strcmp(buffer, Classe_Controllo_Figlio_Cercato))
    {
      // Classe OK: ora controlliamo le coordinate
      GetWindowRect(hwnd, &R_Tmp);

      if (R_Tmp.left == Dati_X_Y_X1_Y1.left
          && R_Tmp.top == Dati_X_Y_X1_Y1.top
          && R_Tmp.right == Dati_X_Y_X1_Y1.right
          && R_Tmp.bottom == Dati_X_Y_X1_Y1.bottom)
        H_Controllo_Figlio_Cercato = hwnd;
    }
  }

  // Se l'handler del controllo figlio e' stato trovato
  // ritorniamo valore ZERO a Windows: questo interrompe
  // il processo di chiamata ciclica della funzione

  return(!H_Controllo_Figlio_Cercato);
}

*)

Function MiaCallBack(Handle:THandle) : boolean ; stdcall;

```

```

var
  WindowText: array[0..255] of char;
  Test_text, Test_Class : Boolean;
Begin
  BH := 0;
  Result := True;
  GetWindowText(handle, WindowText, 255);
  // Verifico se il testo del Bottone è quello prestabilito
  Test_text:= String(WindowText) = BtnText;
  // Verifico se la classe è quella che mi devo aspettare
  FillChar(WindowText, 256, 0); // pulisco il buffer
  GetClassName(handle, WindowText, 254);
  // verifico che la classe sia quella prevista
  Test_Class := String(WindowText) = BtnClass ;

If (Test_text AND (BtnClass = '')) Or
  (Test_Text AND Test_Class AND (BtnClass <> ''))
Then
  Begin
    BH := Handle;
    // Se l'handler del controllo figlio e' stato trovato
    // ritorniamo valore "False" a Windows: questo interrompe
    // il processo di chiamata ciclica della funzione
    Result := False;
  End;
End;

//
// Aziono il Click su un Bottone dell'applicazione "Er Finestra"
// ClassP = Nome della Classe Padre
// WinP = Nome della finestra Padre
// BText = Testo collocato all'interno del bottone da "cliccare"
// La procedura utilizza tre variabili Globali
// BH di tipo THandle : contiene l'Handle dell'oggetto trovato da MiaCallBack
// BTnText di tipo String
// BtnClass di tipo String
//
Function TForm1.Click_Bottone(ClassP, WinP, BText, BClass : string):boolean;
var hp : THandle;

begin
  // BH := 0; // handle Globale settato da MiaCallBack
  BtnText := BText;
  BtnClass := BClass;

  // Handle finestra Padre
  hp := FindWindow(PChar(ClassP), PChar(WinP));
  If hp <> 0 Then EnumChildWindows(hp, @MiaCallBack, LParam(0));
  // Ho trovato Figlio cercato ??
  If BH <> 0 Then
  Begin
    Result := True;
    If IsWindowEnabled(BH) Then
    Begin
      // Ora notificiamo al bottone trovato un click
      SendMessage(BH, WM_LBUTTONDOWN, 0, 0);
    End;
  End;
end;

```

```

    SendMessage(BH, WM_LBUTTONDOWN, 0, 0);
    // minimizzo il bottone (non funziona )
    // SendMessage(BH,SW_MINIMIZE,0,0);
End;
End
Else Result := False;
end;

// Rilevo lo status di un Bottone dell'applicazione "Er Finestra"
// ClassP = Nome della Classe Padre
// WinP = Nome della finestra Padre
// BText = Testo collocato all'interno del bottone da "analizzare"
// La procedura utilizza tre variabili Globali
// BH di tipo THandle : contiene l'Handle dell'oggetto trovato da MiaCallBack
// BTnText di tipo String
// BtnClass di tipo String
// Valori restituiti -1 se bottone non esiste
//          0 se esiste ed è pronto
//          +1 se esiste ma è occupato
Function TForm1.Stat_Bottone(ClassP, WinP, BText, BClass : string):integer;
var hp : THandle;

begin
// BH := 0; // handle Globale settato da MiaCallBack
BtnText := BText;
BtnClass := BClass;

// Handle finestra Padre
hp := FindWindow(PChar(ClassP), PChar(WinP));
If hp <> 0 Then EnumChildWindows(hp,@MiaCallBack,LParam(0));
// Ho trovato Figlio cercato ??
If BH <> 0 Then
Begin
Result := 0 ; // Bottone esistente e pronto
If NOT IsWindowEnabled(BH)
Then Result := 1; // Bottone esistente ma non pronto
End
Else Result := -1; // Bottone inesistente
end;

// Attivazione dei Bottoni sulla Finestra di "Er Finestra"
//
procedure TForm1.Comando_Leggi;
begin
// Eseguo la interpretazione vocale
Click_Bottone('#32770','Er Finestra','Leggi','Button');
end;

procedure TForm1.Comando_Pausa;
begin
// Eseguo la interpretazione vocale
If Not Click_Bottone('#32770','Er Finestra','Pausa','Button')
Then
// Eseguo la interpretazione vocale
Click_Bottone('#32770','Er Finestra','Avvia','Button');
end;

```

```

procedure TForm1.Comando_Esci;
begin
  // Eseguo la interpretazione vocale
  Click_Bottone('#32770','Er Finestra','Esci','Button');
end;

```

```

// Pressione del Pulsante STOP
procedure TForm1.Comando_Stop;
begin
  // Eseguo la interpretazione vocale
  Click_Bottone('#32770','Er Finestra','Stop','Button');
end;

```

(\*\*\*\*\*  
 \*\*\*\*\*

mouse\_event

The mouse\_event function synthesizes mouse motion and button clicks.

VOID mouse\_event(

```

  DWORD dwFlags, // flags specifying various motion/click variants
  DWORD dx,      // horizontal mouse position or position change
  DWORD dy,      // vertical mouse position or position change
  DWORD dwData,  // amount of wheel movement
  DWORD dwExtraInfo // 32 bits of application-defined information
);

```

Parameters

dwFlags

A set of flag bits that specify various aspects of mouse motion and button clicking. The bits in this parameter can be any reasonable combination of the following values:

Value Meaning

**MOUSEEVENTF\_ABSOLUTE** Specifies that the dx and dy parameters contain normalized absolute coordinates. If not set, those parameters contain relative data: the change in position since the last reported position. This flag can be set, or not set, regardless of what kind of mouse or mouse-like device, if any, is connected to the system. For further information about relative mouse motion, see the following Remarks section.

**MOUSEEVENTF\_MOVE** Specifies that movement occurred.

**MOUSEEVENTF\_LEFTDOWN** Specifies that the left button changed to down.

**MOUSEEVENTF\_LEFTUP** Specifies that the left button changed to up.

**MOUSEEVENTF\_RIGHTDOWN** Specifies that the right button changed to down.

**MOUSEEVENTF\_RIGHTUP** Specifies that the right button changed to up.

**MOUSEEVENTF\_MIDDLEDOWN** Specifies that the middle button changed to down.

**MOUSEEVENTF\_MIDDLEUP** Specifies that the middle button changed to up.

**MOUSEEVENTF\_WHEEL** Windows NT only: Specifies that the wheel has been moved, if the mouse has a wheel. The amount of movement is given in dwData

The flag bits that specify mouse button status are set to indicate changes in status, not ongoing conditions. For example, if the left mouse button is pressed and held down, `MOUSEEVENTF_LEFTDOWN` is set when the left button is first pressed, but not for subsequent motions. Similarly, `MOUSEEVENTF_LEFTUP` is set only when the button is first released.

**dx**

Specifies the mouse's absolute position along the x-axis or its amount of motion since the last mouse event was generated, depending on the setting of `MOUSEEVENTF_ABSOLUTE`. Absolute data is given as the mouse's actual x-coordinate; relative data is given as the number of mickeys moved.

**dy**

Specifies the mouse's absolute position along the y-axis or its amount of motion since the last mouse event was generated, depending on the setting of `MOUSEEVENTF_ABSOLUTE`. Absolute data is given as the mouse's actual y-coordinate; relative data is given as the number of mickeys moved.

**dwData**

If `dwFlags` is `MOUSEEVENTF_WHEEL`, then `dwData` specifies the amount of wheel movement. A positive value indicates that the wheel was rotated forward, away from the user; a negative value indicates that the wheel was rotated backward, toward the user. One wheel click is defined as `WHEEL_DELTA`, which is 120. If `dwFlags` is not `MOUSEEVENTF_WHEEL`, then `dwData` should be zero.

**dwExtraInfo**

Specifies an additional 32-bit value associated with the mouse event. An application calls `GetMessageExtraInfo` to obtain this extra information.

**Return Values**

This function has no return value.

**Remarks**

If the mouse has moved, indicated by `MOUSEEVENTF_MOVE` being set, `dx` and `dy` hold information about that motion. The information is given as absolute or relative integer values. If `MOUSEEVENTF_ABSOLUTE` value is specified, `dx` and `dy` contain normalized absolute coordinates between 0 and 65,535. The event procedure maps these coordinates onto the display surface. Coordinate (0,0) maps onto the upper-left corner of the display surface, (65535,65535) maps onto the lower-right corner.

If the `MOUSEEVENTF_ABSOLUTE` value is not specified, `dx` and `dy` specify relative motions from when the last mouse event was generated (the last reported position). Positive values mean the mouse moved right (or down); negative values mean the mouse moved left (or up). Relative mouse motion is subject to the effects of the mouse speed and the two mouse threshold values. In Windows NT, an end user sets these three values with the Mouse Tracking Speed slider of Control Panel's Mouse option; in Windows 95, an end user sets them with the Pointer Speed slider of the

**Control Panel's Mouse property sheet. An application obtains and sets these values with the SystemParametersInfo function.**

The operating system applies two tests to the specified relative mouse motion. If the specified distance along either the x or y axis is greater than the first mouse threshold value, and the mouse speed is not zero, the operating system doubles the distance. If the specified distance along either the x or y axis is greater than the second mouse threshold value, and the mouse speed is equal to two, the operating system doubles the distance that resulted from applying the first threshold test. It is thus possible for the operating system to multiply relatively-specified mouse motion along the x or y axis by up to four times.

The mouse\_event function is used to synthesize mouse events by applications that need to do so. It is also used by applications that need to obtain more information from the mouse than its position and button state. For example, if a tablet manufacturer wants to pass pen-based information to its own applications, it can write a dynamic-link library (DLL) that communicates directly to the tablet hardware, obtains the extra information, and saves it in a queue. The DLL then calls mouse\_event with the standard button and x/y position data, along with, in the dwExtraInfo parameter, some pointer or index to the queued extra information. When the application needs the extra information, it calls the DLL with the pointer or index stored in dwExtraInfo, and the DLL returns the extra information.

See Also

GetMessageExtraInfo, SystemParametersInfo

```
*)  
// Movimenti del Mouse  
(*
```

**Question:**

**How do I use the mouse\_event() procedure to simulate a mouse event?**

**Answer:**

The following example demonstrates using the Windows API function mouse\_event() to simulate a mouse event. When Button2 is clicked, the example code places the mouse over Button1 and clicks it. Mouse Coordinates given are in "Mickey's", where their are 65535 "Mickey's" to a screen's width.

```
procedure TForm1.Button1Click(Sender: TObject);  
begin  
  ShowMessage('Button 1 clicked');  
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);  
var  
  Pt : TPoint;  
begin  
  {Allow Button2 to repaint it's self}  
  Application.ProcessMessages;  
  {Get the point in the center of button 1}  
  Pt.x := Button1.Left + (Button1.Width div 2);  
  Pt.y := Button1.Top + (Button1.Height div 2);
```

```

{Convert Pt to screen coordinates}
Pt := ClientToScreen(Pt);
{Convert Pt to mickeys}
Pt.x := Round(Pt.x * (65535 / Screen.Width));
Pt.y := Round(Pt.y * (65535 / Screen.Height));
{Move the mouse}
Mouse_Event(MOUSEEVENTF_ABSOLUTE or
             MOUSEEVENTF_MOVE,
             Pt.x,
             Pt.y,
             0,
             0);
{Simulate the left mouse button down}
Mouse_Event(MOUSEEVENTF_ABSOLUTE or
             MOUSEEVENTF_LEFTDOWN,
             Pt.x,
             Pt.y,
             0,
             0);;
{Simulate the left mouse button up}
Mouse_Event(MOUSEEVENTF_ABSOLUTE or
             MOUSEEVENTF_LEFTUP,
             Pt.x,
             Pt.y,
             0,
             0);;
end;

*)

//
(* Ultimato il Thread di ricerca effettuato da FileFind
della cartella .\SMS, per prima cosa vengono contati
i messaggi già letti R ed i messaggi da leggere M.
I valori di R e di M vengono comunicati con Er Finestra
*)
procedure TForm1.FileFind1Terminate(Sender: TObject);
var k : integer;
    st : string;
begin
    TL2.Assign(filefind1.FilesFound);
    // FileFind1.ThreadTerminate;

    r := 0;//variabile che conta i messaggi già letti dall'er finestra.
    M := 0;// Messaggi che non sono stati letti.
    For k := 1 to TL2.Count do
        begin
            st := TL2.Strings[k-1];
            st := extractfilename(st);
            if (st[1] = 'R') Then Inc(r)
            else Inc(m);
        end;

    // comunico l'esito della ricerca
    T11.Clear;

```

```

If (r > 1) or (r = 0)
  Then TL1.Add('Trovo : '+IntToStr(r)+' messaggi già letti : ')
  Else TL1.Add('Trovo : '+IntToStr(r)+' messaggio già letto : ');
If (m > 1) or (m = 0)
  Then TL1.Add('e : '+IntToStr(m)+' messaggi da leggere : ')
  Else TL1.Add('e : '+IntToStr(m)+' messaggio da leggere : ');

logf.LogToFile(TimeToStr(now)+' R= '+IntToStr(R)+' - M= '+IntToStr(M));

TrasferimentoMessaggi;

  // Avvio Lettura degli SMS
  w := TL2.Count;
  K := 0;
  timer1.Enabled:=true;

end;

procedure TForm1.Timer1Timer(Sender: TObject);
begin
  if Stato_master
    then LeggiSMS
    else timer3.Enabled := true;
end;

//
// Interpretazione Vocale degli SMS
//
(*
*)
Procedure TForm1.LeggiSMS;
  var
    st, st1, st2 : string;
begin
  // verifico se il Bottone Leggi di Er Finestra è attivo
  // ossia se la lettura del precedente testo si è ultimata
  If (Stat_Bottone('#32770','Er Finestra','Leggi','Button') = 0)
  Then
    Begin
      if (w = 0)
        then
          // Non ci sono files nella cartella .\SMS
          begin
            TL1.Clear;
            timer1.Enabled:=False;
            // attivo il timer di chiusura dell'applicazione
            timer3.Enabled := true;
          end
        else
          // Ci sono files nella cartella .\SMS
          begin
            st := TL2.Strings[k];
            st1 := extractfilename(st); // estraggo nome del file
            st2 := ExtractFilePath(st); // estraggo path
            if (st1[1] <> 'R')
              then

```

```

// Messaggio non ancora letto
begin
  TL1.Clear;
  TL1.LoadFromFile(st);
  TrasferimentoMessaggi;
  // Aggiunta della lettera R in testa al nome
  st2 := st2+'R'+st1;
  If RenameFile(st, st2)
    Then TL2.Strings[k] := st2
    Else logf.LogToFile(TimeToStr(now)+' RENAME FAILED'+ST+'#'+ST2);
  end;
Inc(k);
if k = W // Letti tutti i messaggi ???
then
  Begin
    timer1.Enabled := False;
    k := 0;
    W := 0;
    timer3.Enabled := true;
  End;

end;
end;
End;

//
// Toglie la lettera R all'inizio del nome del file
// rendendo il messaggio leggibile da parte del pgm.
//
procedure TForm1.EliminaR;
  VAR st, st1, st2 : string;
      k : integer;
Begin
  If TL2.Count <> 0
  Then
    Begin
      logf.LogToFile(TimeToStr(now)+' ELIMINO R IN FILENAME');
      For k:= 1 To TL2.Count do
        Begin
          st := TL2.Strings[k-1];
          st1 := extractfilename(st);
          st2 := ExtractFilePath(st);
          If (st1[1] = 'R')
          Then
            Begin
              Delete(st1,1,1);
              st2 := st2+st1;
              RenameFile(st, st2);
              TL2.Strings[k-1] := st2;
            End;
          End;
        End;
      End;
    End;
  End;
end;

//
procedure TForm1.Button2Click(Sender: TObject);

```

```

Begin
  // Toglie la lettera R all'inizio del nome del file
  // rendendo il messaggio leggibile da parte del pgm.
  EliminaR;
end;

procedure TForm1.Timer3Timer(Sender: TObject);
begin
  If (Stat_Bottone('#32770','Er Finestra','Leggi','Button') = 0)
  then
    begin
      PlaySound('Chiusura.wav', 0, SND_ASYNC OR SND_FILENAME);
      Sleep(3000);
      close;
    end;
end;

{Se la funzione mi restituisce il vlaore true
il master è nello stato stand by e posso
continuare mentre se mi restituisce il valore
false il master è nello stato runnig e devo
interrompere}
Function TForm1.Stato_master : boolean;
var
  h, k : HWND;
  s : integer;
  St: array[0..260] of char;
  st1 : string;
begin
  h := FindWindowByTitle('FABRI SOFTWARE');
  k := FindControlByNumber(h,'TButton',1);
  s := GetWindowTextLength(k);
  GetWindowText(k,st,s+1);
  result := (st = 'STAND BY');
end;
end.

```

## **Invio Messaggi SMS**

Questo programma permette di inviare sms, con due modalità d'invio.

Si può mandare un messaggio già precedentemente creato, che deve essere salvato come 'Msg.txt'.

Il file deve contenere nella prima linea il numero di cellulare a cui inviare il messaggio, e dalla seconda linea il testo del messaggio, ricordandosi però di non superare i 160 caratteri.

L'altra modalità d'invio è a testo e destinatario libero; per selezionare quale delle due modalità d'invio utilizzare in alto a destra si trovano due RadioButton.

Nel nostro caso il messaggio viene predisposto dal programma Master; per questo, quando il programma sarà avviato, partirà nella modalità Invio Predefinito.

Per creare questo applicativo abbiamo utilizzato i seguenti oggetti :

- 1 Memo
- 1 Edit
- 1 Button
- 2 RadioButton
- 1 Timer
- VaComm

### **Invio a testo Libero**

In questa modalità d'invio va inserito all'interno dell'Edit il numero del destinatario, mentre nella Memo il testo del messaggio e poi cliccare sul Button 'Invia SMS'. Prima di mandare il messaggio vengono effettuati dei controlli, se il testo è troppo lungo o assente, se il destinatario non è stato immesso.

A questo punto viene inviato un comando al modem : AT+CMGS= ' numero del destinatario '

Si esegue un ritardo (Sleep) di 1 secondo e poi si manda il testo da inviare.

Adesso si attiva il Timer che va a controllare la risposta del modem, in caso di risposta positiva (OK) esegue un file vocale di 'Messaggio inviato correttamente', altrimenti esegue 'Errore durante l'invio', nel caso di risposta negativa (ERROR).

### **Invio Predefinito**

Come già detto prima l'invio predefinito si basa sul file 'Msg.txt'.

All'avvio la prima linea del file viene scritta all'interno della Edit, mentre dalla seconda in poi nella Memo e poi viene inviato il messaggio, come prima descritto.

Quando il programma viene eseguito, passa da due eventi, l'evento OnCreate e quello OnShow

## OnShow

Qui controllo vari parametri, appoggiandomi sul file Setting\_sms.ini

Setto la proprietà PortNum, dell'oggetto VaComm, questo permette al programma di sapere su quale porta com deve interfacciarsi per comunicare con il modem. Poi setto la proprietà Baudrate sempre dell'oggetto VaComm che sarebbe la velocità con la quale il computer comunica con il modem. Controllo inoltre che nel computer sia presente la porta Com tramite la function IsComPortInstalled, nel caso in cui non ci fosse la porta viene emesso un messaggio di errore, mentre se la porta è presente, apre la comunicazione (VaComm1.open).

Per esigenze di programmazione dopo avere fatto questi controlli si eseguono questi comandi:

```
radiobutton1.Checked := true;  
button1.Click;  
timer1.Enabled := true;
```

Setta la modalità d'invio su predefinita  
Sarebbe come cliccare sul bottone Invia SMS  
Abilita il Timer per il controllo

**Il programma in esecuzione si presenta così:**



**unit Unit1;**

**interface**

**uses**

**Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
StdCtrls, VaClasses, VaComm, IniFiles, MMSystem, ExtCtrls;**

**type**

**TForm1 = class(TForm)**

**Edit1: TEdit;**

**Label1: TLabel;**

**Label2: TLabel;**

**Memo2: TMemo;**

**RadioButton1: TRadioButton;**

**RadioButton2: TRadioButton;**

**Button1: TButton;**

**Timer1: TTimer;**

**procedure RadioButton1Click(Sender: TObject);**

**procedure RadioButton2Click(Sender: TObject);**

**procedure Button1Click(Sender: TObject);**

**procedure FormShow(Sender: TObject);**

**procedure VaComm1RxChar(Sender: TObject; Count: integer);**

**procedure Timer1Timer(Sender: TObject);**

**private**

**{ Private declarations }**

**public**

**{ Public declarations }**

**function IsComPortInstalled(port: integer): boolean;**

**procedure Controllo;**

**end;**

**var**

**Form1: TForm1;**

**r, fname: string;**

**b: integer;**

**implementation**

**{ \$R \*.DFM }**

**procedure TForm1.RadioButton1Click(Sender: TObject);**

**var**

**ncount: integer;**

**st: string;**

**tl: TStringList;**

**begin**

```

memo2.Clear;
edit1.Clear;
tl := TStringList.Create;
tl.LoadFromFile('Msg.txt');
ncount := tl.Count;
st := tl.Strings[ncount - 1];
dec(ncount);
memo2.Lines.Add(st);
st := tl.Strings[ncount - 1];
edit1.Text := st;
tl.Free;
end;

```

```

procedure TForm1.RadioButton2Click(Sender: TObject);
begin
    edit1.Enabled := True;
    edit1.Clear;
    memo2.Clear;
end;

```

```

procedure TForm1.Button1Click(Sender: TObject);
var
    dest, testo, st0, st2: string;
begin
    dest := edit1.Text;
    edit1.Enabled := False;
    testo := memo2.Text;
    //memo2.Clear;
    if dest = '' then
        if testo = '' then ShowMessage('Destinatario e Testo non presenti')
        else
            ShowMessage('Destinatario non presente')
    else
        begin
            if testo = '' then ShowMessage('Testo non presente')
            else
                begin
                    if Length(testo) > 160 then
                        begin
                            ShowMessage('Testo troppo lungo (Testo < 160 caratteri)');
                            Testo := '';
                            memo2.Clear;
                        end
                    else
                        begin
                            // Mando il comando invio
                            St0 := 'AT+CMGS=' + dest + '' + Chr(13);
                            VaComm1.WriteText(St0);
                            sleep(1000);
                            VaComm1.WriteText(testo + Chr(26));
                            timer1.Enabled := True;
                        end
                    end
                end
            end
        end
    end
end;

```

```
end;  
end;  
end;  
end;
```

```
function TForm1.IsComPortInstalled(port: integer): boolean;
```

```
var
```

```
MaxPorts: integer;
```

```
hPort: THandle;
```

```
PortNumber: integer;
```

```
PortName, st: string;
```

```
begin
```

```
st := '';
```

```
{ Verifica della piattaforma }
```

```
case Win32Platform of
```

```
VER_PLATFORM_WIN32_NT: MaxPorts := 256;
```

```
VER_PLATFORM_WIN32_WINDOWS: MaxPorts := 9;
```

```
end;
```

```
for PortNumber := 1 to MaxPorts do
```

```
begin
```

```
if PortNumber > 9 then
```

```
PortName := '\\.\COM' + IntToStr(PortNumber) // ask Microsoft why...
```

```
else
```

```
PortName := 'COM' + IntToStr(PortNumber);
```

```
hPort := CreateFile(PChar(PortName),
```

```
GENERIC_READ or GENERIC_WRITE,
```

```
0, nil, OPEN_EXISTING, 0, 0);
```

```
// note that ports already in use by other apps
```

```
// will *NOT* be detected here
```

```
if not (hPort = INVALID_HANDLE_VALUE) then st := st + PortName + ',';
```

```
CloseHandle(hPort);
```

```
end;
```

```
Result := Pos('COM' + IntToStr(port), st) <> 0;
```

```
end;
```

```
procedure TForm1.FormShow(Sender: TObject);
```

```
var
```

```
speed, n: integer;
```

```
z: Tinifile;
```

```
begin
```

```
b := 0;
```

```
memo2.Clear;
```

```

edit1.Clear;
fname := ExtractFilePath(Application.ExeName);
n := length(fname);
fname := copy(fname, 1, n - 6) + 'Setting_sms.ini';
Z := TiniFile.Create(fname);
VaComm1.PortNum := Z.ReadInteger('MODEM', 'COMPORT', 0);
speed := Z.ReadInteger('MODEM', 'SPEED', 9600);
case speed of
  9600: VaComm1.Baudrate := br9600;
  19200: VaComm1.Baudrate := br19200;
end;
Z.Free;

// Verificare Presenza della porta Seriale
if not IsComPortInstalled(VaComm1.PortNum) then
begin
  ShowMessage('Porta COM non Trovata');
  Close;
end
else
begin
  //showmessage('Porta COM Trovata');
  VaComm1.Open;
end;
end;

procedure TForm1.VaComm1RxChar(Sender: TObject; Count: integer);
begin
  r := r + VaComm1.ReadText;
  if (Pos('OK', r) <> 0) or
    (Pos('ERROR', r) <> 0) or
    (Pos('>', r) <> 0) then
  begin
    Inc(b);
    memo2.Lines.Add(R);
    r := '';
  end;
end;

procedure TForm1.Controllo;
var
  St3: string;
  f, e: integer;
begin
  f := memo2.Lines.Count;
  repeat
    if pos('OK', memo2.Lines[f]) <> 0 then
      begin
        playSound('Msg_Inv_Ok.wav', 0, SND_ASYNC or SND_FILENAME);
        sleep(5000);
      end;
  until f = 0;
end;

```

```

    e := 1;
    f := 0;
end
else
    dec(f);
until f = 0;
if e = 0 then
begin
    repeat
        if pos('OK', memo2.Lines[f]) <> 0 then
            begin
                playSound('Msg_Inv_ERROR.wav', 0, SND_ASYNC or SND_FILENAME);
                sleep(5000);
                f := 0;
            end
        else
            dec(f);
        until f = 0;
    end;
end;
end;

```

```

procedure TForm1.Timer1Timer(Sender: TObject);
begin
    controllo;
end;

end.

```

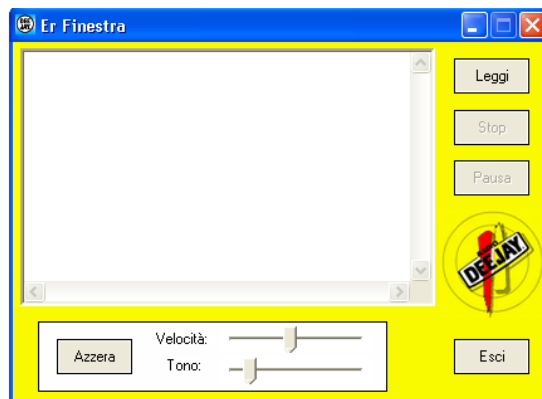
---

### Sintetizzatore VOCALE:

---

Programma ER FINESTRA:

Questo programma funziona come uno “text speak” , cioè si limita a convertire il testo in audio. Infatti, funziona indipendentemente dalla sintesi vocale installata nel computer e si limita soltanto a convertire in voce ciò che sta scritto.





ISTITUTO PROFESSIONALE DI STATO  
PER L'INDUSTRIA E ARTIGIANATO  
- MORETTO-

TESINA D' ESAME DI STATO

# Tesina di Maturità

## Inglese

### Inglese

COMPUTER NETWORKS  
WANS AND LANS

BORRA ALESSANDRO  
DA SILVA CARLOS ANDRE  
CLASSE 5° BZ

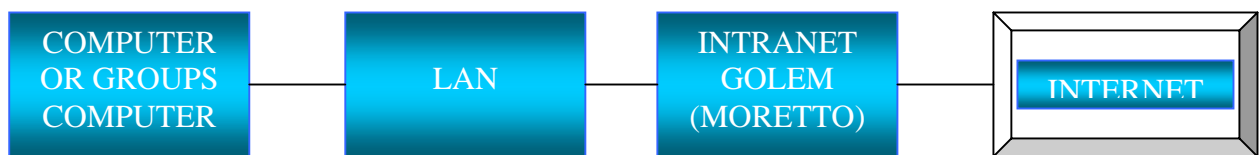
ANNO SCOLASTICO 2004/05

- Computer Network
- LAN
- Protocol
- Intranet
- Internet

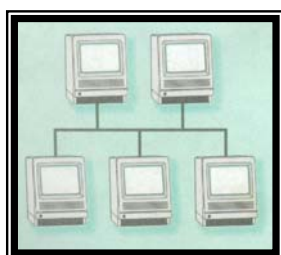
A computer network is a group of computers which are interconnected by telephone lines or other high speed communications links or other high-speed communication links in order to **share information's** and processing resources. Networks enable data to be transferred rapidly from one computer to another.

On the internet is called TRASMISSION CONTROL PROTOCOL / Internet Protocol (TCP/IP), a standard procedure for regulating data transmissions between computers.

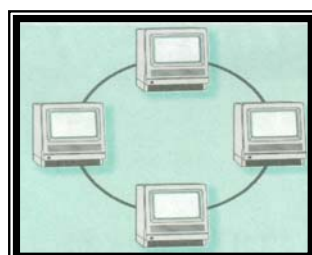
Two basic network types local – area networks (lan), and wide area networks (wan). Lan connected computers and peripheral devices in a limited physical area, such as a business office, laboratory, or college campus. A typical lan consists of two or more personal computers, printers, ecc. The internet is the largest network of all, but private networks are also used by governments, banks and other organizations in order, for example, to access confidential databases or transfer funds.



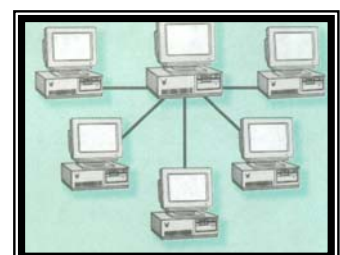
Three network types:



BUS



RING



STAR

ISTITUTO PROFESSIONALE DI STATO  
PER L'INDUSTRIA E ARTIGIANATO  
- MORETTO -

TESINA D' ESAME DI STATO

# Tesina di Maturità

## STORIA

*Gli Armeni un popolo dimenticato*

BORRA ALESSANDRO  
DA SILVA CARLOS ANDRE  
CLASSE 5° BZ – SETTORE ELETTRONICO

**ANNO SCOLASTICO 2004/05**

## Repubblica Armena



**Superficie:** 29.808 Km<sup>2</sup>  
**Abitanti:** 3.336.000 (stime 2001)  
**Densità:** 112 ab/Km<sup>2</sup>

**Forma di governo:** Repubblica presidenziale  
**Capitale:** Erevan (1.200.000 ab.)  
**Altre città:** Vanadzor 147.000 ab., Gyumri 125.000 ab.  
**Gruppi etnici:** Armeni 93%, Russi 2%, altri 5%  
**Paesi confinanti:** Georgia a NORD, Azerbaigian a EST e a SUD, Iran a SUD, Turchia ad OVEST

**Monti principali:** Aragats Lerr 4090 m  
**Fiumi principali:** Araks 1070 Km (totale, compresi tratti azero, iraniano e turco), Kasah, Debed  
**Laghi principali:** Lago Sevan 1400 Km<sup>2</sup>  
**Isole principali:** -  
**Clima:** Continentale

**Lingua:** Armeno (ufficiale), Russo  
**Religione:** Armena Apostolica  
**Moneta:** Dram armeno

*"Chi sono gli Armeni?"*

## La religione.

Com'è riportato nella Storia di Agatangelo, conservata in armeno e nelle versioni greche ed araba, il popolo armeno fu il primo popolo convertitosi al Cristianesimo ad opera del Patriarca Gregorio l'Illuminatore (Lusavuric) che nel 301 battezzò il re degli armeni Tiridate. Una tradizione orale riferisce che gli evangelizzatori di questo popolo furono S.Bartolomeo e S.Taddeo. La religione fu influenzata notevolmente dagli avvenimenti storici. Fu determinante infatti per il Concilio di Calcedonia la mancata presenza della chiesa armena dovuta alla guerra in atto tra persiani ed armeni. I dettami del Concilio furono riportati agli armeni dai siriaci, che traducendogli li avevano falsati. Questo portò molti vescovi a rompere con Roma e furono conseguentemente accusati di essere monofisiti. Altri vescovi, accortisi di ciò, non aderirono allo scisma e rimasero in perfetta comunione con Roma. I primi diedero vita alla Chiesa Apostolica Armena, i secondi a quella Cattolica.

Determinante è l'opera della Chiesa per il popolo armeno. Padre Adonyan sostiene infatti che essere armeno vuol dire essere cristiano. L' Armenia costituisce una goccia cristiana in un oceano mussulmano.

## "SE I NOSTRI FIGLI SI DIMENTICANO DI NOI, TU , O MONDO, RICORDA LORO CHE..."

Nel 1914 la situazione armena peggiora irrimediabilmente. In quell'anno infatti il governo turco decide di entrare in guerra a fianco degli imperi centrali e subito si lancia alla conquista dei territori azeri "*irredenti*". La Terza Armata turca, impreparata, male equipaggiata, mandata allo sbaraglio in condizioni climatiche ostili, viene presto sbaragliata a Sarikamish nel gennaio 1915 dalle forze sovietiche. L'esercito turco indica i responsabili della disfatta negli armeni che, allo scoppio della guerra avevano comunque assicurato il proprio sostegno all'impresa turca. Il clima si fa sempre più teso e, tra il dicembre del '14 ed il febbraio del '15, il **Comitato Centrale del partito Unione e Progresso**, diretto dai medici **Nazim** e **Behaeddine Chakir**, decide la soppressione totale degli armeni. Vengono creati speciali battaglioni irregolari, detti **tchéte**, in cui militano molti detenuti comuni appositamente liberati; essi hanno addirittura autorità sui governi ed i prefetti locali e quindi godono di un potere pressoché assoluto.

L'eliminazione sistematica prende l'avvio nel 1915, quando i battaglioni regolari armeni vengono disarmati, riuniti in gruppi di lavoro ed eliminati di nascosto. Il piano turco, pensato e diretto dal **Ministro dell'Interno Talaat**, prosegue poi con la soppressione della comunità di Costantinopoli ed in particolare della ricca ed operosa borghesia armena: tra il 24, che resta a segnare la data commemorativa del genocidio, ed il 25 aprile, 2345 notabili armeni vengono arrestati mentre tra il maggio ed il luglio del 1915 gli armeni delle province orientali di Erzerum, Bitlis, Van, Diyarbakir, Trebisonda, Sivas e Kharput vengono sterminati. Solo i residenti della provincia di Van riescono a riparare in Russia grazie ad una provvidenziale avanzata dell'esercito sovietico. Nelle città viene diffuso un bando che intima alla popolazione armena di prepararsi per essere deportata; si formano così grandi colonne nelle quali gli uomini validi vengono raggruppati, portati al di fuori delle città e qui sterminati. Il resto della popolazione viene indirizzato verso Aleppo ma la città verrà raggiunta solo da pochi superstiti: i nomadi curdi, l'ostilità della popolazione turca, i tchéte e le inumane condizioni a cui sono sottoposti fanno sì che i deportati periscano in gran numero lungo il cammino. Dopo la conclusione delle operazioni neppure un armeno era rimasto in vita in queste province.

La seconda parte del piano prevedeva il genocidio della popolazione armena restante, sparsa su tutto il resto del territorio. Tra l'agosto del 1915 ed il luglio del 1916 gli armeni catturati vengono riuniti in carovane e, malgrado le condizioni inumane cui vengono costretti, riescono a raggiungere quasi integre **Aleppo** mentre un'altra parte di deportati viene diretta verso **Deir es-Zor**, in Mesopotamia. Lungo il cammino, i prigionieri, lasciati senza cibo, acqua e scorta, muoiono a migliaia. Per i pochi sopravvissuti la sorte non sarà migliore: periranno di stenti nel deserto o bruciati vivi rinchiusi in caverne.

A queste atrocità scamperanno solo gli armeni di Costantinopoli, vicini alle ambasciate europee, quelli di Smirne, protetti dal generale tedesco Liman Von Sanders, gli armeni del Libano e quelli palestinesi. Il consuntivo numerico di questo piano criminale risulta alla fine:

- **da 1.000.000 a 1.500.000 di armeni vengono eliminati nelle maniere più atroci. In pratica i due terzi della popolazione armena residente nell'Impero Ottomano è stata soppressa e, regioni per millenni abitate da armeni, non vedranno più, in futuro, nemmeno uno di essi.**

- **circa 100.000 bambini vengono prelevati da famiglie turche o curde e da esse allevati smarrendo così la propria fede e la propria lingua.**

• **considerando tutti gli armeni scampati al massacro il loro numero non supera le 600.000.**

Su tutte valga la testimonianza del Console italiano **Giovanni Gorrini** che così scrisse: "Dal 24 giugno non ho più dormito ne mangiato. Ero preso da crisi di nervi e da nausea al tormento di dover assistere all'esecuzione di massa di quegli innocenti ed inermi persone. Le crudeli cacce all'uomo, le centinaia di cadaveri sulle strade, le donne ed i bambini caricati a bordo delle navi e poi fatti annegare, le deportazioni nel deserto: questi sono i ricordi che mi tormentano l'anima e quasi fanno perdere la ragione." Anche l'intervento della Santa Sede tramite il Papa Benedetto XV non produsse alcun effetto, in funzione anche del fatto che **i turchi avevano proclamato la guerra santa**. Successivamente, approfittando degli sconvolgimenti in corso in Russia a causa della rivoluzione, gli armeni sotto il controllo dell'impero zarista si ribellano e, il 28 maggio 1918, dichiarano la propria indipendenza. In seguito, dopo la presa di alcuni territori nell'Armenia turca, verrà proclamata la nascita della **Repubblica Armena**. Durante i lavori del **Trattato di Sevrès** venne perfino riconosciuta l'indipendenza al popolo armeno e la sua sovranità su gran parte dei territori dell'Armenia storica ma, come altre volte in futuro, tutto resterà solo sulla carta. Infatti il successivo **Trattato di Losanna** (1923) annullerà il precedente negando al popolo armeno persino il riconoscimento della sua stessa esistenza.

La caduta del regime turco alla fine della Grande Guerra e la seguente ascesa alla guida del paese di **Kemal Atatürk** non cambiò la situazione. Infatti, tra il 1920 ed il 1922, con l'attacco alla Cilicia armena ed il **Massacro di Smirne**, il nuovo governo portò a compimento il genocidio. Dopo questi ultimi crimini non un solo armeno vivo lasciò traccia in Turchia.

---

#### LA STORIA DI ARMIN T. WEGNER

---

Armin T. Wegner, la cui collezione fotografica documenta le condizioni dei campi di deportazione armeni nel 1915-1916, è nato in Germania nel 1886. Quando scoppiò la Prima Guerra Mondiale, si arruolò come infermiere volontario in Polonia durante l'inverno del 1914-1915, e fu decorato con la Croce di Ferro per aver assistito i feriti in combattimento. Nell'Aprile del 1915, in seguito all'alleanza militare fra Germani e Turchia, fu mandato nel Medio Oriente come membro del Corpo tedesco.

Tra luglio ed agosto, usò il suo mandato per indagare sulle voci che erano arrivate fino a lui da diverse fonti circa i massacri degli armeni. Nell'autunno dello stesso anno, con il grado di sottotenente al seguito del feldmaresciallo Van der Goltz, comandante del 6° esercito ottomano in Turchia, viaggiò per l'Asia Minore. Eludendo gli ordini rigorosi delle autorità turche e tedesche (che avevano lo scopo di evitare



la diffusione di notizie, informazioni, corrispondenza, testimonianze oculari), Wegner raccolse appunti, annotazioni, documenti, lettere e scattò centinaia di fotografie nei campi di deportazione armeni. Con l'aiuto di consolati ed ambasciate di paesi stranieri, fu in grado di spedire parte del suo materiale in Germania e negli Stati Uniti. I suoi canali clandestini per la corrispondenza furono scoperti e Wegner fu arrestato dai tedeschi su richiesta del comando turco, e fu messo a servire nei reparti affetti da colera. Essendosi ammalato gravemente, lasciò Bagdad diretto a Costantinopoli nel novembre del 1916. Nella cintura aveva nascosto le lastre fotografiche sue e di altri ufficiali tedeschi con immagini del genocidio degli Armeni di cui era stato testimone. A dicembre di quello stesso anno fu richiamato in Germania.

Wegner era molto toccato dalla tragedia del popolo armeno di cui era stato testimone oculare nella Turchia ottomana. Tra il 1918 e il 1921 diventò il membro attivo di movimenti pacifisti ed antimilitari. Nel 1968 ricevette un invito a recarsi in Armenia da parte dei Catholicos di tutti gli

Armeni e fu decorato con l'ordine di San GRECOGIO L'Illuminatore . Armin Wegner morì a Roma all'età di 92 anni il 17 maggio 1978.



***Due giorni dopo il massacro del 30 ottobre 1895 a Erzerum: fossa per seppellire le vittime armene.***



***Una donna armena e i suoi bambini durante la deportazione (foto di Armin Wagner).  
Fotografie tratte dal volume "Breve storia del genocidio degli armeni" di Claude Mutafian e Metz Yegh rn.***

### **Fatto Politico:**

Ancor oggi la Turchia con l'appoggio di USA ed Israele cerca di negare o di minimizzare le cifre e la drammaticità dell'olocausto armeno, sovvenzionando anche pseudo storici di varie nazionalità. Troppe sono, però, le testimonianze non solo armene ma di diplomatici e militari Inglesi, Americani, Italiani e di altre nazionalità che documentarono il genocidio avendone avuta diretta coscienza. Fra di loro spicca l'ufficiale tedesco Armin Wagner, che a rischio della vita, essendo un militare di un Paese alleato della Turchia nella I Guerra Mondiale, documentò anche fotograficamente le indicibili sofferenze di questo popolo martire.

Il negazionismo turco poggia su tre argomenti.

- Il primo, sostenuto sin dal 1915, cerca di ribaltare la responsabilità dei tragici eventi accusando gli Armeni di aver tradito la Turchia tentando una specie di genocidio antiturco. Si prendono a tal fine ad esempio alcuni attacchi a villaggi turchi da parte di bande armene venute dalla Russia.
- La seconda tesi cerca di negare la premeditazione dello sterminio ma dall'esame di atti estratti nel 1988 dagli archivi ottomani emerge sostanzialmente solo una grossolana falsificazione di documenti.
- Il terzo argomento verte sulle statistiche. Secondo il patriarcato armeno di Costantinopoli vivevano nell'Impero Ottomano 2.100.000 persone, i turchi dicono 1.290.000, quindi per i primi le vittime sarebbero un milione e mezzo per i secondi fra 200.000 ed 800.000.

Perché furono sterminati fra il 1915 ed il 1923 un milione e mezzo di Armeni di Turchia? Abbiamo provato a schematizzarne alcuni motivi fra i più degni di nota con la premessa che nell'Impero Ottomano, erede "diretto ed ultimo" dell'Impero Romano, grande realtà multiculturale, multi-etnica e multiconfessionale, gli Armeni furono rispettati ed in un certo senso anche protetti, ma considerati sempre oltre che minoranza etnico-linguistica-religiosa, una realtà temibile economicamente e politicamente inaffidabile o comunque sospetta.

Questa è sinteticamente la ragione dei massacri hamidiani ossia del tragico precedente dell'olocausto del 1915-1923. Alla fine dell'Ottocento, infatti, fra 50.000 e 300.000 Armeni furono sterminati perché ritenuti eversivi nei confronti del potere ottomano già in profonda ed inarrestabile crisi. Dati precisi sui morti non ce ne sono, stante l'assenza di testimoni attendibili, neutrali.

Già alla fine dell'Ottocento, peraltro, l'influenza delle idee di rinascita nazionalista avevano permeato la ricca borghesia armena dove fiorivano le logge massoniche che tanta importanza avranno anche nella genesi del movimento dei giovani turchi e nella ideologia modernizzatrice di Mustafà Kemal "Ataturk".

Gli Armeni costituivano per il Sultano, così come per chi lo detronizzò nel 1908, una pericolosa enclave cristiana nel cuore della madrepatria turca, non cioè una realtà periferica come Serbi, Bulgari, Greci.

Altra grave "colpa" ... era di possedere una cultura ed una lingua scritta antichissime in conflitto pertanto con il concetto base del Panturchismo: la lingua turco-turanica è la

lingua più antica dell'Umanità intera da cui, poi, sono derivate tutte le altre ... ad essa quindi debitrice ...

Come accennato gli Armeni, minoranza coesa, erano in media nettamente più colti, ricchi e socialmente influenti rispetto ai Turchi ed alle altre nazionalità dell' Impero e poi dello Stato Laico turco.

Altra causa scatenante furono i profughi mussulmani sia balcanici, ossia Bosniaci e Bulgari , sia provenienti dal Caucaso i quali , a seguito delle sconfitte subite dai Turchi, venivano insediati nelle regioni della Anatolia orientale tradizionalmente abitate in larghissima maggioranza da Armeni (Costituenti pertanto l' Armenia occidentale dell' ipotetico Stato Gran – Armeno nel sogno ottocentesco di rinascita nazionale armena , elemento, però , puramente culturale ).

L' Impero Ottomano era pertanto vicino al collasso, anche in Nordafrica aveva perso la Libia ceduta all' Italia.

Gli Armeni sono stati qui di un capro espiatorio. Neavevano pienamente le caratteristiche, purtroppo ...

L' olocausto armeno , quindi , è la risultante di un insieme di sfortunate coincidenze ed accadimenti, naturale epilogo della diffusione di idee quali quella che coincidendo territorio ed etnia si debba ...procedere... ad estirpare tutto ciò che in un modo o nell' altro è alieno !

E' nostra opinione pertanto che né la Fede Islamica in quanto tale né tanto meno motivi di carattere ideologico-politico possano aver prodotto un crimine di tale portata.

Se si considera che comunque in Turchia si celebrarono già nel 1919 alcuni processi a carico dei responsabili del massacro degli Armeni, è davvero singolare che dopo novant' anni la Turchia non si assuma la piena responsabilità di questo odioso crimine, cercando , ad esempio , di individuare con i moderni sistemi aereofotogrammetrici le centinaia di fosse comuni ancora ignote ma sicuramente presenti nella odierna Turchia orientale ed in Siria.

Come ha scritto lo storico turco-americano Teren Ashcian la perdita della memoria di questo genocidio è un grave danno per gli stessi Turchi in considerazione , anche , aggiungiamo noi, del prossimo ed inevitabile ingresso della Turchia nella Unione Europea, patria comune – speriamo - di libertà, fraternità ed uguaglianza .

All'entrata si oppongono i partiti armeni, ma ogni sforzo è vano. I giovani turchi iniziano la loro follia e per gli armeni inizia il METZ YEGHERN (IL GRANDE MALE). Con questo nome gli armeni chiamano il loro genocidio, un grande male è infatti quello che sente l'armeno per questa violenza che ha subito e subisce ancora.

### **In sei mesi i turchi uccideranno da un milione e mezzo a due milioni di Armeni.**

#### **▪ Modalità dello sterminio**

In primo luogo voglio precisare che tutta l'operazione viene mascherata come un'azione di spostamento di persone da ipotetiche zone di guerra. Tutto ciò perché i giovani turchi vorrebbero far credere che la sparizione di due milioni di persone sia dovuta al caso. Mi accingo ora ad illustrare le modalità di sterminio:

- eliminazione del cervello della nazione. Il 24 Aprile 1915 vengono arrestati gli esponenti dell'élite culturale armena. Tra questi c'è anche Krikor Zohrab, deputato del parlamento, che pensava di essere amico di Talaat. Questi intellettuali saranno deportati all'interno dell'Anatolia e massacrati. Ci vorranno cinquant'anni per ricostruire una classe pensante.
- Eliminazione della forza. Gli Armeni dai 18 ai 60 anni vengono chiamati alle armi a causa della guerra in atto. Questi ,da bravi cittadini ,si arruolano. Un decreto

stabilisce il disarmo di tutti i militari armeni, che vengono costituiti in battaglioni del genio. A gruppi di 100 verranno isolati e massacrati. Di 350.000 soldati armeni nessuno si salverà.

- E' il turno di donne vecchi e bambini. I medici Nazim e Behaeddin Chackir sguinzagliano la loro organizzazione segreta. Nei luoghi vicino al mare si procede all'annegamento noyades. Lo sterminio diretto viene applicato anche nelle zone in cui incombeva l'avanzata russa per il timore che alcuni si potessero salvare. Veniamo ora alle deportazioni (tehcir ve taktil = deportazione e massacro).
- Il capo della gendarmeria locale dà ordine ai maschi armeni di presentarsi al comune, appena arrivati vengono imprigionati ed eliminati fuori dal villaggio. Si incomincia la deportazione con la scusa dello spostamento da zona di operazioni belliche. L'editto dovrebbe essere comunicato con cinque giorni d'anticipo, contrariamente si dà molto meno tempo per non offrire loro la possibilità di prepararsi. Fuori dal villaggio intanto aspettano curdi e turchi per impadronirsi della abitazioni. Con una legge del 10.6.1915 e tutta un'altra serie di leggi, che si seguono, i beni della perone deportate vengono dichiarati "beni abbandonati" emvali metruke" quindi soggetti a confisca e riallocazione. Allontanatisi i convogli, questi sono privati dei carri (bisogna camminare) si possono così facilmente eliminare le persone per fatica senza dover usare proiettili.

Ho espressamente citato ed utilizzato, per la ricostruzione degli eventi, materiale tedesco (ricordo che i tedeschi erano alleati dei turchi), americano (l'america era neutrale) in maniera di non poter essere accusato di aver avvalso di fonti faziose e non fedeli perché viziate da azione di propaganda bellica antiturca..

Voglio precisare che i turchi usarono l'Islam, ma che non fu uno sterminio di religione, infatti molto la nazione armena deve agli arabi (disprezzati dai turchi che li considerano una razza inferiore) mussulmani del vicino oriente, che hanno accolto i superstiti dimostrando loro benevolenza, a volte salvandoli direttamente.

I turchi non sono ancora paghi delle loro efferatezze. Per dar forma al sogno panturco bisogna eliminare tutti gli Armeni, anche gli orientali. I russi, a causa della rivoluzione d'ottobre si ritirano, i turchi attaccano ma una straordinaria mobilitazione di popolo impedisce questa ulteriore tragedia.

Mustafà Kemal detto Atatürk (padre dei turchi), fondatore della Turchia moderna, basata sul mito della razza e della forza militare, dà il via alla terza fase del genocidio.

Atatürk si accorda con i bolscevichi per la spartizione della repubblica armena, che si era formata alla fine della guerra; è un altro massacro. Atatürk attacca anche la Grecia. L'incendio di Smirne del 1921 segna la fine dell'oriente greco. Atatürk attacca anche la Cilicia, sotto mandato francese, i francesi sono costretti ad abbandonarla, è la fine del sogno di uno stato armeno di Cilicia. Si viene ad interrompere una storia di civiltà e cultura che durava da più di tremila anni. Nessuna nazione europea interviene.

Questo è un genocidio che non solo è rimasto impunito ma anche premiato, infatti la maggior parte del territorio nazionale armeno fa ora parte della Turchia. L'armeno vive ora, da più di ottant'anni, nella più grande diaspora conosciuta dall' evo moderno.

## ***L'Armenia attuale***

---

Durante e dopo l'attuazione del piano criminale turco gran parte degli scampati e dei sopravvissuti furono costretti all'esilio ed alla diaspora. Nel 1991 a seguito della dissoluzione dell'URSS è nata la **Repubblica Armena** sulle ceneri dell'ex Repubblica Sovietica Armena. **Il 90%**

**dell'Armenia storica, comunque rimane sotto il controllo della Turchia** che, oltre a non voler ammettere alcuna responsabilità riguardo al genocidio, rifiuta categoricamente la restituzione anche parziale dei territori da loro occupati. Nel 1989 scoppia la guerra con il vicino Azerbaigian per il controllo dell'**Artzak** (Nagorno-Karabach) l'enclave armena in territorio azero, che sembra essersi recentemente concluso con la conquista dell'indipendenza della provincia armena.



Recentemente i rapporti tra Curdi ed Armeni sono migliorati in seguito alle persecuzioni turche che hanno colpito entrambi i popoli ma il governo di Ankara si ostina ancora a non voler riaprire la frontiera kurdo-armena. Inoltre i rapporti tra l'Armenia e l'Azerbaigian turcofono sono tuttora tesi a causa delle rivendicazioni azere sul territorio del neonato stato di Artzak e per le rivendicazioni armene sul **Nakitcevan** provincia affidata all'Azerbaigian dal Trattato russo-turco del 1921, area che taglia i rapporti diretti tra lo Stato di Armenia e la **provincia armena di Tabriz** in territorio iraniano.

## ***Il riconoscimento del Genocidio da parte della Comunità Internazionale***

---

Attualmente il genocidio armeno è stato riconosciuto come realtà storica di cui la Turchia dovrà farsi carico in diverse sedi. L'ONU, anche se in sordina, lo ha fatto il 29 agosto del 1985 mentre il **Parlamento Europeo** si pronunciò in proposito il 18 giugno 1997. Tra le nazione attivatesi in questo senso tra le prime è stato l'**Uruguay** ed alcuni stati degli USA (**Massachusetts, California, New Jersey, New York, Wisconsin, Pennsylvania, Rhode Island, Virginia ed Illinois** in ordine di tempo a partire dal 1978 al 1995) mentre ne il Governo statunitense, ne il Consiglio di Stato hanno preso iniziative simili. Anche la **Duma della Federazione Russa** ha ufficialmente riconosciuto quanto accaduto agli armeni. Per quanto riguarda l'Italia sono state prese iniziative a livello comunale quali quelle di Milano, nel novembre '97, e recentemente di Roma. Inoltre per il giorno 31/3/2000 è stata posta all'ordine del giorno della Camera una mozione, presentata già nel '98 dall'onorevole G. Pagliarini (Lega Nord per l'Indipendenza della Padania) e sottoscritta da 165 deputati di vari partiti, che mira al riconoscimento, da parte del Governo Italiano, del genocidio armeno. Dopo lunga attesa questa mozione è stata accantonata dall'attuale maggioranza (governo Amato) definendo il momento storico-politico non opportuno per approvare il documento.

# RIFLESSIONE

---

A tutt'oggi il riconoscimento del genocidio da parte della comunità internazionale sembra ancora ben lontano dall'essere una realtà ed i timidi tentativi, quali quello dell'Assemblea Nazionale Francese, di dare dignità storica ai fatti avvenuti in quegli anni sono stati tutti immediatamente insabbiati dalle inconsulte reazioni turche e dal vergognoso silenzio-assenso delle grandi potenze, primi fra tutti gli USA, che hanno sempre dato maggiore importanza ai propri interessi politici ed economici piuttosto che alla giustizia ed al rispetto di quei principi morali ai cui spesso loro stessi fanno appello e di cui si sentono custodi.

---

<http://www.comunitaarmena.it> è il nuovo sito della **comunità armena di Roma** molto ben fatto ed utile per seguire le attività proposte dalla comunità.

<http://www.wavefront.com/~homelands/armenia.html> il sito è completamente in inglese ed è una pagina generale dove si possono trovare molti link con varie informazioni sulla storia e cultura armena, gruppi di discussione, gli eventi politici od il sito del governo armeno.

<http://www.genocide.am/> è un sito completamente in inglese che oltre a fornire i dati sulla tragedia armena invita materialmente al sostegno di una massiccia campagna per il riconoscimento del genocidio.

<http://www.calpoly.edu/~pkiziria/pub-files/FACTS.html> il testo, disponibile anche in italiano, fornisce un resoconto completo degli avvenimenti dell'epoca del genocidio e numerose foto originali.

<http://www.agora.stm.it/politic/nagorno-karabach.html> dal sito generale di Agorà è possibile visitare quello del movimento indipendentista armeno nell'enclave azera dell'**Artsak (Nagorno Karabach)**. Tra i link visitabili anche quello sul conflitto con la repubblica azera che pare finalmente, da poco, giunto ad una conclusione.

[http://members.xoom.it/Voce\\_Armena/index.htm](http://members.xoom.it/Voce_Armena/index.htm) a questo indirizzo è possibile accedere alla versione in lingua italiana del periodico armeno **Voce Armena**.

<http://space.tin.it/edicola/vshahbaz> un bellissimo sito che tratta nello specifico la storia dell'attuale Repubblica autonoma dell'**Artsak**, meglio nota in Italia con il nome turco di **Nagorno Karabagh**.

<http://web.infinito.it/utenti/a/armpap> è l'indirizzo del sito personale del signor **Armenag Papazian** armeno della comunità milanese. Sito molto completo in cui si possono trovare molte informazioni su tutti gli aspetti della cultura armena ma anche sulle associazioni delle varie comunità.

**BIBLIOGRAFIA :**

[www.google.com](http://www.google.com)

[www.ipsiamoretto.com/golem](http://www.ipsiamoretto.com/golem)

*Materiale didattico concesso dal golem*

*Manuale Pocket Digicom*

**RINGRAZIAMENTI:**

*Ringraziamo la disponibilità del gruppo insegnanti della classe 5° BZ, che ci hanno aiutato a preparare questa tesina, in particolare il Prof. Azzani Cleto che con la sua disponibilità, siamo riusciti a concludere questo ambizioso progetto.*

**SI RINGRAZIA INOLTRE :**

*PROF. BERTANZA ENRICO*

*PROF. PRANDELLI FABIO*

*PROF. VITONOFRJO DE TRIZIO*