

Script Lua per il controllo remoto di droni

Tesi sperimentale

Marco Sgobino

Università degli Studi di Udine



In questo elaborato di tesi sperimentale sono stati effettuati:

- ① lo studio della documentazione e comprensione del funzionamento di OpenTX
- ② il progetto e sviluppo di script in Lua eseguiti su firmware OpenTX
- ③ la sperimentazione tramite simulatore e successivamente con radiocomando reale

Introduzione a OpenTX

OpenTX è un firmware per radiocomandi open source, rilasciato in licenza *GNU GPLv2*.

Esso è gestito e sviluppato dalla comunità di sviluppatori e piloti, in collaborazione con le aziende interessate.

Il modello a sorgente libero consente ad esperti, sviluppatori e piloti di godere di una maggiore sicurezza, stabilità e controllo sul software, nonché allo studente di esaminarne liberamente il funzionamento ed apprenderne i meccanismi interni.



Scritto in C++, supporta oltre 20 radiocomandi. I modelli utilizzati sono:

- il *Taranis FrSky QX7S*



Scritto in C++, supporta oltre 20 radiocomandi. I modelli utilizzati sono:

- il *RadioMaster TX16S*



Il firmware OpenTX è composto da due distinti applicativi:

- il *System Firmware*, responsabile della gestione degli input, della lettura delle impostazioni, della produzione di immagini su schermo
- il *Transmitter Firmware*, si occupa della trasmissione radio

La memorizzazione avviene attraverso una memoria a stato solido. Lo schermo consente all'utente la configurazione delle impostazioni, la lettura della telemetria (RSSI, GPS, batteria residua del radiocomando).

Ogni telecomando è in grado di riprodurre annunci vocali.

I mixer consentono di elaborare uno o più input mediante coefficienti, offset, o funzioni più complesse, e direzionarlo in uno o più canali d'uscita (fino a 32).

Esistono più tipi di mixer:

peso ed offset l'output è ottenuto mediante la formula

$$output = sorgente \cdot peso + offset$$

diff riduce l'effetto di un input in una direzione e l'amplifica nella direzione opposta

function applica una funzione matematica o disuguaglianza

expo applica un andamento esponenziale

curve applica una curva personalizzata (fino a 17 punti selezionabili)

OpenTX supporta gli script in Lua dalla versione 2.0.

Il firmware consente l'esecuzione di script con regole e struttura precisa.

Esistono due categorie principali di script:

- gli script monouso
- gli script persistenti

I primi sono caricati all'avvio del firmware o del modello in memoria, i secondi vengono evocati una singola volta ed interrompono gli altri script.

Script persistenti

Sono caricati all'avvio, ed eseguiti fino allo spegnimento del radiocomando o alla selezione di un nuovo modello.

È possibile eseguirne fino ad un massimo di 7 contemporaneamente.

Vi sono più tipi di script persistenti:

script di modello svolgono funzioni analoghe ai mixes, con possibilità arbitrarie di elaborazione

script di funzione del tutto simili agli script di modello, attivabili con un pulsante o finché una determinata condizione risulta vera

script di telemetria presentano dati e immagini al pilota, e possono riprodurre annunci vocali

widget hanno funzionalità paragonabili agli script di telemetria. Sono supportati dai modelli più avanzati, è possibile personalizzarne dimensione e colori da radiocomando



Gli script monouso sono evocati dal menù del radiocomando, o da una specifica funzione.

Interrompono tutti gli altri script, la loro esecuzione è ripristinata al termine dello script monouso.

Sono l'ideale per realizzare programmi come ad esempio i wizard di installazione.

Script di telemetria

Gli script di telemetria seguono una precisa struttura, e vi sono 3 tipi di funzioni principali:

funzione `init()` viene eseguita all'avvio dello script (opzionale)

funzione `run()` viene eseguita periodicamente, quando la schermata di telemetria è visibile

funzione `background()` viene eseguita periodicamente a prescindere dalla schermata mostrata

Uno statement di `return` a fine script collega le definizioni delle funzioni, che possono avere un nome diverso, con le funzioni principali richieste da OpenTX.

I widget sono l'evoluzione degli script di telemetria.

Nella prima parte:

`stringa nome` il nome del widget, massimo 10 caratteri

`table options` contiene le opzioni configurabili dal pilota tramite interfaccia grafica

`funzione create()` viene chiamata all'avvio del widget

`funzione update()` viene chiamata quando si verifica una modifica alle opzioni

Nella tabella delle opzioni possono essere presenti i seguenti elementi:

VALUE valore numerico costante

SOURCE sorgente di input da cui prelevare il valore

BOOL variabile booleana costante

COLOR colore espresso in formato RGB565

Essi sono modificabili dall'utente da radiocomando, mediante le opzioni del widget.

Le funzioni principali dei widget sono:

funzione `refresh()` eseguita periodicamente quando lo schermo è visibile

funzione `background()` eseguita periodicamente quando lo schermo non è visibile

Anche in questo caso uno statement di `return` collega le definizioni delle funzioni con le funzioni principali previste da OpenTX.

Si occupano della stampa su schermo:

- `lcd.clear()` permette di azzerare la schermata, obbligatorio negli script di telemetria
- la stampa avviene con `lcd.drawText()` oppure `lcd.drawTimer()`
- `lcd.setColor()` permette di impostare un colore personalizzato al testo

Lo schermo del *RadioMaster* supporta una modalità a colori con risoluzione 480×272 , mentre lo schermo del *Taranis* supporta soltanto la modalità bianco e nero con risoluzione 128×64 .

OpenTX Companion è un software messo a disposizione dagli sviluppatori di OpenTX per la configurazione del radiocomando interamente da computer desktop.

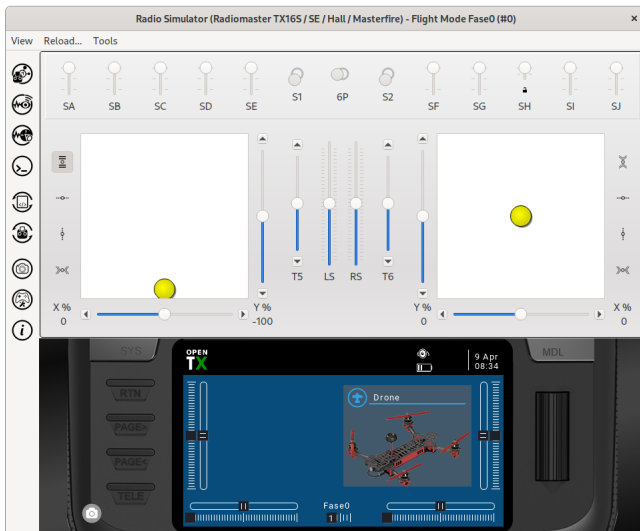
Esso consente:

- il caricamento del firmware nel radiocomando e la sincronizzazione della scheda SD
- la configurazione del radiocomando
- la creazione e configurazione dei modelli
- l'esecuzione del backup delle impostazioni

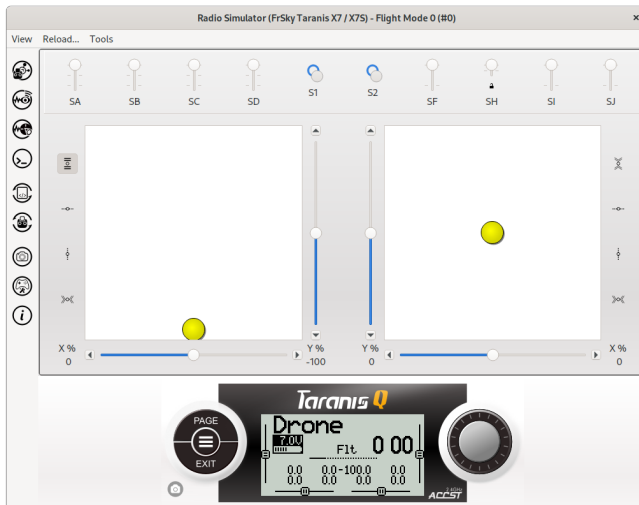
OpenTX Simulator è il secondo software a disposizione degli sviluppatori. Esso è in grado di condurre accurate simulazioni del radiocomando con un alto grado di fedeltà.

Sono supportati oltre 20 radiocomandi differenti.

Aspetto della simulazione del radiocomando *RadioMaster TX16S*



Aspetto della simulazione del radiocomando *Taranis QX7S*



In questa tesi sperimentale sono stati realizzati due script Lua:

- lo script per il cronometro vocale, `TmrCnt`
- lo script per il conto alla rovescia vocale, `CntDwn`

Ciascuno script è stato realizzato in due versioni, una per radiocomando in esame.

Lo script TmrCnt realizza un timer su schermo, con lo scopo di tener conto del tempo trascorso dall'inizio della fase di volo.

Oltre che visualizzare il tempo nella schermata del radiocomando, le informazioni del timer sono riprodotte da annunci vocali.

Questo approccio consente al pilota di mantenere il contatto visivo con il drone, senza la necessità di dover visualizzare l'informazione sul radiocomando.

Lo script fa uso di un timer interno al radiocomando per il conteggio del tempo trascorso.

Le opzioni dello script presentate al pilota sono:

Timer il timer scelto, con possibili valori 1, 2 e 3

Alert il numero di secondi che intercorrono fra ciascun annuncio vocale

Color il colore del testo (per i radiocomandi che supportano i widget)

Invers flag booleano per l'inversione dei colori (negli schermi in bianco e nero)

Le due funzioni principali, `refresh()` e `background()` vedono i calcoli suddivisi in ulteriori funzioni, `playTimer()` e la funzione responsabile della stampa della telemetria, `drawTelemetry()`.



playTimer()

La funzione fa uso di un flag booleano, inizialmente al valore `false`, per il controllo della riproduzione dell'annuncio vocale.

A seguito della riproduzione dell'annuncio, ulteriori riproduzioni indesiderate sono evitate dal flag al valore `true`.

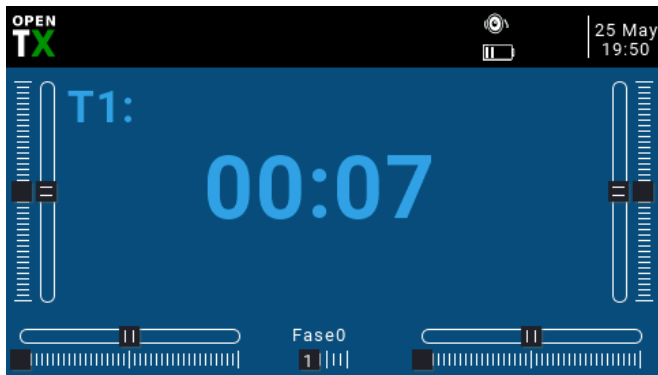
Quando si verificano le condizioni idonee, il flag è riportato al valore `false`.

La lettura del numero avviene mediante la funzione interna `playDuration()`.

La stampa è realizzata con le seguenti modalità:

- tenendo conto delle possibili dimensioni dei widget nella versione per telecomandi che supportano i widget
- sfruttando in ogni caso l'intero spazio disponibile, anche per gli schermi con caratteristiche inferiori e non in grado di supportare i widget
- per una migliore ottimizzazione, la funzione di stampa è chiamata solo ed esclusivamente tramite la funzione `refresh()`, attiva nel caso sia visibile la schermata di telemetria

Lo script si presenta infine con il seguente aspetto (versione per il *TX16S*):



Lo script si presenta infine con il seguente aspetto (versione per il QX75):



Lo script CntDwn realizza un conto alla rovescia a partire da un valore impostabile dall'utente (valore predefinito 180 secondi), con precisi annunci vocali.

Gli annunci vocali vengono riprodotti ogni minuto, e negli ultimi 30, 15 secondi, ed infine da 10 secondi fino allo scadere del tempo residuo. Gli annunci vocali consentono al pilota di non interrompere il contatto visivo con il drone.

La struttura dello script è del tutto simile alla precedente, eccezion fatta per la funzione di riproduzione degli annunci, maggiormente complessa.

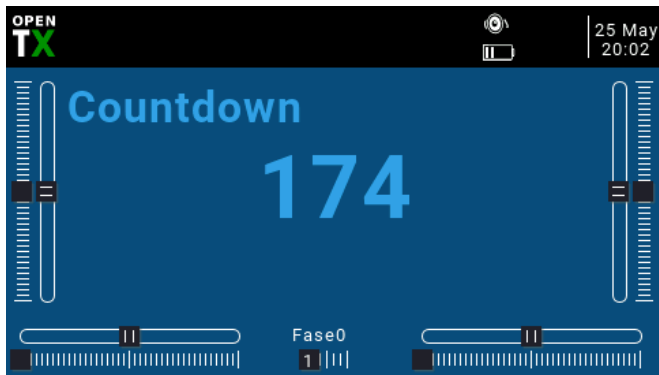
La funzione dedicata alla riproduzione degli annunci è stata progettata per riprodurre annunci a determinati secondi, e fa ancora uso di un flag booleano di controllo.

La principale differenza è che a causa della maggiore complessità il ripristino del flag avviene in una funzione esterna, `resetPlayedFlag()`.

Tale funzione si occupa di valutare i casi specifici per cui il flag di controllo debba essere ripristinato al valore `false`.

Il ripristino infatti avviene sempre al secondo successivo all'annuncio vocale, così da evitare possibili ripetizioni indesiderate.

Lo script si presenta infine con il seguente aspetto (versione per il *TX16S*):



Lo script si presenta infine con il seguente aspetto (versione per il QX7S):



Lo studio della documentazione, la scrittura degli script e l'analisi del loro comportamento tramite il simulatore è potuta per intero avvenire con mezzi esclusivamente software.

Ogni script è stato ugualmente testato su radiocomandi reali, ottenendo il medesimo comportamento riscontrato durante la fase di test tramite simulatore.

In definitiva, è stato possibile creare due script di telemetria, in grado di supportare radiocomandi dalle caratteristiche assai differenti e con numerose possibilità di espansione futura.

Alcuni spunti per la realizzazione di script futuri potrebbero essere:

- ottenimento ed elaborazione di ulteriori sorgenti, fra cui
 - RSSI intensità del segnale in ricezione
 - GPS coordinate GPS ed indicazione dell'ultima posizione nota in caso di segnale assente
 - RxBt tensione della batteria del modello
- produzione di avvisi critici (scarso RSSI, raggiungimento di una determinata area o coordinata GPS, tensione di batteria del modello minore di una certa soglia)
- controllo diretto del drone, sempre prestando attenzione alla sicurezza e ad eventualità criticità in tale ambito

GRAZIE PER
L'ATTENZIONE

