BVE4
GUIDA
ALL'UTILIZZO DEI SEGNALI

Si vuole con questa guida dare una indicazione di base su come costruire e gestire i segnali per BVE4. Infatti, mentre la gestione dei segnali era totalmente automatica nella versione BVE2, e molto giapponese per dirla tutta, la versione BVE4 modifica ed innova rispetto alla precedente e permette una più completa personalizzazione degli stessi.[1]

Non credo siano disponibili altre guide sui segnali nella versione di BVE4 e quindi si tratta di un lavoro sperimentale che spero possa comunque interessare chi volesse programmare linee di BVE "intelligenti"
Naturalmente per ogni consiglio, guida o suggerimento il sito principe di riferimento è quello di Luigi Cartello (http://bve.altervista.org) e quello di Trenomania (http:www.trenomania.org)

Chiunque volesse contribuire ad impreziosire la guida od a correggere eventuali errori, può contattarmi tramite il forum BVE (http://www.trenomania.org - forum BVE nickname: Like)

In modo assolutamente generico e per rendere l'idea del presente lavoro possiamo definire il segnale gestito da BVE4 come una sequenza di diapositive che appaiono in uno schermo: forse il concetto è un po' semplicistico, ma rende sicuramente l'idea. Il tempo si scorrimento delle diapositive, cioè dei segnali, è gestito dal simulatore in almeno 2 modi che andremo a trattare approfonditamente di seguito.

1. DEFINIZIONI

Sono le istruzioni necessarie per gestire questa guida; ne indichiamo la sintassi, procedendo più in là a dare istruzioni sull'utilizzo.

1.1. IN GENERALE

$Rnd(x1;x2)
Questa istruzione ad ogni caricamento della linea genera un numero intero casuale, compreso fra un minimo ed un massimo, in qualsiasi punto della *Track Section (*ndr *Sezione Linea)*.
x1 = numero intero minimo da cui partire nella generazione casuale
x2 = numero intero massimo a cui arrivare nella generazione casuale

1.2. SEZIONE PERCORSO - ROUTE SECTION

Route. Signal (x1) x2 inserisce la velocità massima per i semafori inseriti nella linea
x1= numero indice del segnale che viene utilizzato nella definizione dell'oggetto signal nella Structure Section: x1 può assumere ora numero intero fra 0 e 255.
x2= massima velocità in Km/h permessa dopo aver superato il segnale abbinato al numero x1.

Route.Interval x1 attiva il "treno fantasma"
x1 = numero di secondi prima dell'orario di partenza della stazione in cui parte il treno fantasma.

[1] **Nota Bene**: le linee scritte in BVE4 differiscono da quelle scritte in BVE2 specie nell'uso dei segnali. Il simulatore, all'atto di caricamento delle linee di formato diverso restituirà errori di lettura del file.

Se x1 = 0 il treno fantasma è disattivato e tutti i segnali sulla linea sono verdi, salvo quello di partenza alla stazione (vedere il comando .STA nella Track.Section)

1.3. SEZIONE STRUTTURA - STRUCTURE SECTION

.Beacon (x1) x2 (x1 da 0 a 255)
x1 = numero indice del relè o magnete creato
x2 = file del segnale creato.

.Signal (x1).Load x2; x3 (x1 da 0 a 255)
prevede la creazione di un segnale personalizzato con la relativa combinazione di luci
x1 = numero indice del segnale creato
x2 = file del segnale creato.
x3 = file delle luci che vengono abbinate al segnale.

1.4. SEZIONE LINEA FERROVIARIA - TRACK SECTION

.Section x1; x2; x3; xn.....
x0 = numero indice 0 della luce del segnale che viene adottato nel comando .SigF al parametro x1,
x1 = numero indice 1 della luce del segnale che viene adottato nel comando .SigF al parametro x1,
x2 = numero indice 2 della luce del segnale che viene adottato nel comando .SigF al parametro x1,
xn = numero indice n della luce del segnale che viene adottato nel comando .SigF al parametro x1,

Non c'è limite al valore n salvo il numero di luci previsti dal segnale.

.SigF x1; x2; x3; x4
x1 = numero indice del segnale
x2 = sezione che il segnale presidia o controlla 0 = stessa sezione di linea creata dal comando .section, 1 = sezione immediatamente successiva.
x3 = numero segnato che indica i metri di distanza (orizzontale) dal binario base (binario 0)
x4 = numero segnato che indica i metri di altezza (verticale) dal binario base (binario 0)

Non è più possibile creare il palo in automatico. Il comando va obbligatoriamente usato e collegato comando .section.

.Beacon x1; x2; x3; x4
E' il nuovo comando che colloca relé o altro dispositivo di sicurezza lungo i binari che invia informazioni alla CAB del treno. È in più usato in modo non standard dalla .dll di Oskari S.
x1 = tipologia del .beacon
x2 = numero indice del beacon
x3 = sezione che il beacon presidia o controlla 0 = stessa sezione, 1 = sezione successiva.
x4 = numero intero arbitrario inviato all'ATS plug-in della CAB del treno.

.Pretrain x1
Indica il momento in cui è passato in quel punto il treno precedente
x1 = ora, minuti e secondi di passaggio del treno precedente nel formato HHMMSS0

N.B. Il presente oggetto è incompatibile con l'oggetto Route.Interval e può essere ripetutamente usato sulla linea.

2. LA COSTRUZIONE DI UN SEGNALE

Come anticipato nella premessa, il segnale ora è una istruzione personalizzata. Pertanto ogni linea ferroviaria del mondo potrà avere il proprio sistema di segnalazione, con qualche limite all'utilizzo.

La costruzione di un segnale è una operazione che avviene in due fasi.

2.1. Creazione della struttura (o oggetto sfondo del segnale).

Deve essere creato un file .x contenente semplicemente le coordinate dello sfondo, senza immagini. Il file può essere ottenuto con un tipo .b3d o .csv e convertito mediante l'opzione File – exportX del programma Structureviewer.
Questo file di struttura può essere creato definitivamente alle coordinate x e y scelte (z invece va posta sempre alla stessa distanza), oppure può essere spostato successivamente collocando il segnale mediante il comando sigf.
Facendo un esempio:
coordinate x = –0.35 +0.35 = 70 cm di larghezza
coordinate y = 0.00 +0.70 = 70 cm di altezza
coordinate z = 0.00 = profondità zero.
In questo modo si crea un oggetto sfondo di dimensioni 70cm X 70 cm.

Il file può avere qualsiasi nome, ma tale nome vincolerà successivamente il nome da dare alle immagini del segnale.
Facendo un esempio il file si potrebbe chiamare: Protez2v.x = segnale di protezione a 2 vele.

2.2. Creazione del segnale (immagine)

Deve essere creata ora una immagine del segnale nella stessa cartella in cui è stato salvato il file di sfondo .x. e deve essere un'immagine .bmp con larghezza e lunghezza pari ad un multiplo di 2 pixel (es. 32*64 pixel). L'immagine può essere di qualsiasi risoluzione o formato di colore, mentre il colore di trasparenza va rigorosamente nero.

Il nome del file deve essere identico a quello della struttura dello sfondo con aggiunta di un numero intero che va da **zero** a **n** (dove n è un numero naturale intero).

Nel nostro esempio precedente, ecco creati alcuni files che chiamerò:

 Protez2v0.bmp  Protez2v2.bmp  Protez2v4.bmp

Ogni immagine conterrà il segnale da proiettare. Nell'esempio sono stati creati 3 files immagini con l'aggiunta di:

- 0 che rappresenta il rosso
- 2 che rappresenta il giallo
- 4 che rappresenta il verde

E' appena ovvio che ognuno può costruire una sequenza di immagini che vuole. La sequenza proposta negli esempi è stata tratta dagli oggetti nella cartella "Segnali" della linea Sanbie.

Per arricchire l'immagine del segnale creato è possibile aggiungere un palo o altro supporto che però deve essere rigorosamente un file oggetto .obj; il simulatore BVE4 non genererà in automatico alcun palo contrariamente alla versione BVE2.

3. DEFINIZIONE DI UN SEGNALE

Lo sfondo e le relative immagini vanno ora trattate nel file di route. Facendo uso delle definizioni del capitolo 1 provvediamo ora ad inserire le istruzioni.

Innanzitutto definiamo I limiti di velocità imposti dai segnali che abbiamo creato e li collochiamo nella sezione Route all'inizio del file

Es:
Route.Signal(0) 0
Route.Signal(2) 100
Route.Signal(4) 125

Successivamente dobbiamo enunciare nella sezione struttura quale tipo di segnale andremmo a utilizzare nella linea:

Es:
With structure
…
Signal(1).Load sanaro\segnali\Protez2v; sanaro\segnali\Protez2v

Faccio notare che non ci sono le estensioni dei files, ma il primo è un .x ed il secondo è una serie di .bmp; inoltre le dichiarazioni dei files .bmp non hanno l'estensione numerica (nell'esempio 0 2 e 4) che è stata omessa.

Dopo queste enunciazioni, inserite nell'esempio, avremo in sostanza un segnale (il tipo 1) da usare in linea che potrà presentare 3 immagini o colori:

- Rosso con limite zero km/h
- Giallo con limite 100 km/h
- Verde con limite 125 km/h

In questa fase possiamo anche definire un relè o beacon che descriveremo successivamente:
.Beacon(1) sanaro\segnali\magnete.x

L'estensione del file .x per il beacon è obbligatoria e l'oggetto creato con .x è un oggetto in tre dimensioni e non solo uno sfondo come l'oggetto segnale.

4. COLLOCAZIONE DEL SEGNALE

Il segnale sulla linea presidia un tratto della linea ferroviaria. La linea ferroviaria viene suddivisa in tratti più o meno lunghi ognuno controllato da un segnale.
Il comando che colloca il segnale è .Sigf ed il tratto di linea si definisce usando il suo secondo parametro;
Il semaforo potrà assumere i colori definiti dai numeri dei files .bmp creati prima.

Nella sezione linea ferroviaria, o Track Section, alla progressiva prestabilita inseriremo il segnale che abbiamo definito in precedenza.

Le istruzioni utilizzabili sono le seguenti;

Es:
1000,.Section 0; 4, ; il segnale assumerà i colori 0=rosso e 4 = verde
.SigF 1; 1; -2.5; 5 ; il segnale è di tipo 1, presidia il tratto 1000-1999 m ed è posto a –2.5m a sinistra e 5 metri in altezza. Il tratto è creato perché il secondo parametro è 1.
2000,.Section 0; 2 ; 4, ; il segnale assumerà i colori 0=rosso, 2= giallo e 4 = verde
.SigF 1; 1; -2.5; 5 il segnale è di tipo 1, presidia il tratto 2000-in poi ed è posto a –2.5m a sinistra e 5 metri in altezza. Il tratto è creato perché il secondo parametro è 1.

Attenzione alle seguenti osservazioni:

- il primo segnale collocato sulla linea non può avere come secondo parametro 0 (zero) il simulatore darebbe un errore irreversibile.
- La sequenza dei semafori non è nuovamente fissa, ma definita dal comando .section.
- Il limite di velocità nel tratto di linea presieduto dal segnale è stabilito dal comando route.signal definito prima.
- Da esperimenti fatti, il secondo parametro 0 che significa che il segnale presiede lo stesso tratto di linea, può essere usato solo se il successivo segnale è dello stesso tipo del precedente, cioè il primo parametro (tipo di segnale) è uguale al successivo.

Se prima del segnale si vuole passare una informazione in cabina si deve utilizzare una nuova istruzione: .beacon.
Il "*beacon*" è un relé o altro dispositivo magnetico che, posto fra i binari, si attiva al passaggio del treno, dando informazioni nella CAB al macchinista od al presidio ferroviario che controlla la linea.
In BVE4, tale istruzione passa un valore alla cabina del treno affinché, prima della visione del segnale, il macchinista possa avere informazioni sullo stato del segnale stesso, o il treno sia frenato se vengono violati alcuni parametri di velocità.
In Italia questo tipo di segnalazione è detta "ripetizione di segnale in cabina", nelle linee inglesi è detto AWS ed attiva un dispositivo sonoro e visivo nelle cabine dei treni; inoltre viene pure abbinato ad un dispositivo di sicurezza detto TPWS che ferma il treno in caso di violazione di velocità o di superamento del magnete con segnale di via impedita (rosso).

A questo punto è doveroso fare un accenno alla .dll di Oskari Saarekas creata per le CAB evolute di BVE4 che prevede l'utilizzo di alcuni comandi .beacon in modo particolare che inviano istruzioni automatiche alla CAB per il tipo di segnalazione AWS e TPWS. Ne descrivo uno solo, il primo.

Traducendo dal file Beacons.html di Oskari Sarekas (http://koti.mbnet.fi/lopomo/trainsoft/):
"Il comando beacon ha quattro parametri: il primo numero intero è il tipo di beacon, che si distingue da altri beacons. Il secondo numero intero è l'oggetto indice per visualizzare l'oggetto piazzato nel luogo del beacon. Per i beacon generici è possibile che tu voglia creare un oggetto invisibile. Il terzo numero intero unisce il beacon ad un specifico segnale. Per ultimo il quarto numero intero è un parametro supplementare passato al plug-in. Se viene usato un parametro di velocità, devono essere usati kilometri all'ora.
Descrizione:
.beacon 44000; x; 1; 0, magnete di avvicinamento del segnale AWS. Un allarme AWS si attiva se il segnale non è di aspetto 'chiaro' (verde) nel momento in cui il treno passa sopra il magnete."

Nel nostro esempio quindi aggiungeremo:
800,.Beacon 44000;1;1;0,
1800, .Beacon 44000;1;1;0,
In questo modo, prima di impegnare il segnale posto alla progressiva 1000, viene passato in cabina il colore del segnale che presidia la tratta 1000-1999.

Da notare il secondo parametro che indica il numero indice del segnale ed il terzo parametro = 1 che indica che il beacon rileva il semaforo del tratto successivo presidiato dal segnale.

Purtroppo le CAB di treni italiani (e non solo) di BVE4 non sono state ancora implementate per la gestione del ripetizione di segnale in cabina.

5. USO DEL SEGNALE

Non ci resta che attivare dinamicamente il segnale che altrimenti sarebbe sempre verde.
Per poterlo attivare sono disponibili due istruzioni

5.1. Il comando INTERVAL

Il concetto è abbastanza semplice. Il comando fa partire un treno "fantasma" un numero di secondi definito dal comando. Il treno fantasma parte ed impegna la linea prima del tuo treno e fa scattare tutti i segnali al passaggio.
Alle fermate alle stazioni si ferma esattamente 5 metri davanti al punto di fermata e riparte esattamente il numero di secondi indicato dal comando.
Es.
Route.Interval 23
Significa che il treno "fantasma" partirà 23 secondi prima del tuo treno.

Se fra il tuo treno ed il treno fantasma la distanza è di almeno due settori di linea, il segnale si avvicinerà al numero più alto (nel nostro esempio è il 4 = verde);
Se è a meno di due settori il segnale si avvicinerà allo zero;
Se il treno fantasma è nel settore successivo a quello impegnato dal tuo treno, il segnale sarà quello dell'immagine 0 (zero, nel nostro esempio 0 = rosso).

La velocità del treno fantasma è pari al rapporto fra la distanza da percorrere fra due stazioni e la differenza fra l'orario di partenza e quello di arrivo: la velocità del treno fantasma non varia mai

nemmeno se la linea fosse particolarmente tortuosa od in salita. La velocità massima che il treno fantasma raggiungerà nel simulatore BVE4 è di circa 80-90 Km/h., valore verificato con varie prove empiriche.
Questo fa si che tale comando sia più adatto a linee metropolitane.

Si potrebbe arricchire ulteriormente la casualità dell'istruzione utilizzando il comando $Rnd cioè il generatore Random di numeri casuali che restituirà un numero di secondi sempre diverso al comando .Interval

Es.
Route.Interval $Rnd(15;25)
Significa che il treno fantasma partirà ad ogni avvio della route un numero di secondi casuale fra 15 e 25.

5.2. Il comando PRETRAIN

Trattasi di una istruzione più sofisticata e NON compatibile con l'istruzione precedente. Mentre l'istruzione Route.Interval è posto all'inizio della route, il comando pretrain è posto nella sezione della linea ferroviaria o track section. Anche questo tipo di istruzione si basa sulla percorrenza nella linea di un treno "fantasma", ma in modo differente e può quindi essere utilizzato per tratte con una velocità di crociera superiore a 90 Km/h.

Il comando .Pretrain può essere collocato liberamente sulla linea, meglio alla medesima progressiva dei segnali.
Ad ogni istruzione incontrata il simulatore confronta l'orario attuale del tuo treno e la progressiva kilometrica della linea sulla quale è presente il comando .pretrain successivo con orario uguale o inferiore.
Se ci sono due o più sezioni di differenza presidiate da un segnale allora il segnale si pone sul verde o sull'ultima immagine posta dal comando .section, (nel nostro esempio 4= verde) se invece c'è un'unica sezione di linea di differenza, il segnale di protezione della linea successiva si pone sul rosso o sulla prima luce posta dal comando .section, (nel nostro esempio 0 = rosso) o ad una luce intermedia in base alla sequenza di luci prevista dal comando .section e gestite dal segnale. In questo modo si crea un continuo confronto fra i vari comandi .pretrain posti sulla linea e l'orario che ha il treno, ignorando quindi la velocità con cui il tuo treno sta percorrendo la linea. L'uso sapiente del comando .pretrain permette una velocità di crociera più o meno elevata indipendentemente dai limiti di velocità imposti sulla linea.

Es.
Il tuo treno è a 500 metri alle ore 8:5605
Sulla linea sono presenti i seguenti comandi (le istruzioni .section e .signal sono state omesse per semplicità):
500,.Pretrain 085500
1000,.Pretrain 085530
2000,.Pretrain 085600
3000,.Pretrain 085630
4000,.Pretrain 085700
In questo caso alle 08:5605 il treno fantasma è a oltre 2000 metri per cui ci sono ben 2 sezioni di differenza; il segnale posto a 500 sarà verde.

Se il tuo treno accelera ed alle 08:5649 è a 2900 metri, allora il treno fantasma sta impegnando invece il tratto fino a 4000 e pertanto c'è una sola sezione di differenza; il segnale a 3000 sarà sicuramente rosso.

Il comando può essere ulteriormente sofisticato aggiungendo il generatore di numeri casuali ad uno degli elementi del comando.

Es.
2000,.Pretrain 0856 $Rnd(15;25)
Nell'esempio il treno fantasma è passato ad un orario casuale che va dalle 08:56 e 15 secondi e 08:56 e 25 secondi.

L'uso sapiente di questo generatore casuale potrà variare i segnali sulla linea e dare una simulazione sempre diversa; i segnali sicuramente non saranno più tutti verdi!

6. LIMITI DEI SEGNALI IN BVE4

E' doveroso provare a definire cosa non possono fare i segnali di BVE4 o non sono riuscito a sperimentare a fondo.

- Non possono lampeggiare e questo può essere un limite per quelli italiani
- Non ritengo si possa facilmente gestire un segnale di avviso abbinato con uno di blocco perché i segnali cambiano e pertanto se un segnale di avviso precedente poteva indicare un successivo limite, per effetto del comando .pretrain il segnale di blocco potrebbe scattare e porsi sul via libera, vanificando il segnale di avviso precedente. Forse si potrebbe risolvere ponendo il secondo parametro del segnale di avviso a zero e quindi vincolarlo con quello successivo, ma per il momento non sono riuscito a sperimentarlo a fondo.

Spero con questa piccola guida aver stuzzicato la curiosità e la voglia di gestire i segnali in modo dinamico.
Mi scuso con chi non è italiano, ma per il momento ho scritto la presente guida nella lingua a me più naturale: se qualcuno vuole cimentarsi a tradurla lo autorizzo fin da ora.

BUON LAVORO!

LUCA REVELLO
Biella, Italia.