Guida su come configurare EMC2

Dino Del Favero <dino@mesina.net,dino@delfavero.it>

v0.1.9 06/04/2007

Estratto

Questo documento è stato redatto per facilitare la comprensione e la configurazione del software EMC2

Avviso di Copyright

Indice

Capitolo 1

Premessa

In questo tutorial mi riferirò alla versione di EMC sucessiva alla 2.1.0 dato che nel passaggio dalla versione 2.0.x alla 2.1.x vi sono stati dei *lievi* cambiamenti nei file di configurazione.

Capitolo 2

Cos' è e a cosa serve EMC2

EMC (Enhanced Machine Control) è un software che permette di trasformare un normale personal computer in un vero e proprio controllo numerico real-time per macchine CNC.

EMC si basa sul sistema operativo GNU/Linux e da questo ne eredita stabilità, affidabilità, precisione e sicurezza. Precisione e sicurezza sono garantite dal kernel real-time (il kernel è la parte di codice addetta al controllo di tutte le periferiche collegate al computer).

Capitolo 3

Cosa serve per utilizzare EMC2

Per prima cosa un personal computer della famiglia i386, attualmente non esistono porting per hardware non i386.

Una distribuzione GNU/Linux. Per avere un sistema stabile e performante vi consiglio di utilizzare Ubuntu come distribuzione GNU/Linux, infatti per questa distribuzione è possibile scaricare un cdrom *live*, che quindi non necessita di installazione almeno per le prime prove, installare il sistema partendo dal sistema *live* è cosa molto intuitiva e semplice, basta seguire le indicazioni fornite dallo script di installazione. Se non avete già il cdrom lo potete creare seguendo le istruzioni presenti alla pagina internet visibile all'indirizzo `http://www.linuxcnc.org/content/view/21/4/lang,it/`

Opzionale, ma consigliato, un' elettronica di interfacciamento possibilmente optoisolata da collegare al PC tramite la porta parallela, un' elettronica *di potenza* e dei motori passo-passo. Una macchina su cui montare i motori passo-passo.

Capitolo 4

Cosa compone EMC2

Non ancora redatto

Capitolo 5

Iniziamo a configurare EMC2

5.1 Dove si trovano i file di configurazione

In questa guida assumo come distribuzione **Ubuntu 6.06 Dapper Drake LTS.**

Tutti i file di configurazione di EMC2 si trovano in sotto-directory all'interno del path "/etc/emc2/sample-configs", vi sono parecchie directory all'interno di queso path perché EMC2 supporta molti hardware differenti; Ad EMC2 è possibile collegare macchine CNC con diversi tipi di azionamenti, motori passo-passo con o senza encoder, servomotori DC con encoder, servomotori AC con encoder, tramite molti tipi di schede dedicate e molte *porte* differenti (porta parallela, scheda multiIO su bus PCI o su bus ISA, schede servo control, ecc.)

Per ora in questo documento tratterò solamente la configurazione per macchine con **motori passo-passo collegati al PC tramite la porta parallela e controllati tramite i segnali si *passo* e *direzione*** pertanto ci interesserà solo il contenuto della directory "/etc/emc2/sample-configs/stepper".

In questa directory ci sono i sequenti file:

- README
- stepper_mm.ini
- stepper_inch.ini
- sim_inch.ini
- standard_pinout.hal
- sim_pinout.hal
- xylotex_pinout.hal
- core_stepper.hal
- stepper.tbl

- stepper.var
- emc.nml

Nel file README è contenuta una breve descrizione della configurazione, questo file non è strettamente necessario.

Vi sono tre file ”INI”, questi file contengono la configurazione generale della macchina.

Vi sono alcuni file ”HAL” che descrivono come sono collegati i vari componenti della macchina.

Vi sono poi i file stepper.tbl, stepper.var ed emc.nml che servono ad EMC2 per salvare dei dati importanti al buon funzionamento del sistema.

5.2 Com’ è composto un file ”INI”

Un file ”INI” è composto di varie sezioni, individuare una sezione è molto semplice dato che ogniuna inizia con un etichetta racchiusa tra parentesi quadre [COME_QUESTA].

In un file ”INI” possono essere incluse varie sezioni tra cui:

EMC contiene informazioni generali

DISPLAY setta alcune caratteristiche dell’interfaccia utente

TASK setta il tipo di controllore

RS274NGC setta il file contenente le informazioni specifiche dell’interprete dei comandi

EMCMOT setta il modulo del movimento e alcune caratteristiche del controllo

HAL setta il tipo di hardware

TRAJ setta i parametri del modulo di pianificazione della traiettoria

AXIS_0 . . . AXIS_n] setta i parametri di ogni asse

EMCIO setta le variabili di input e output

Naturalmente nei file ”INI” possono esserci anche commenti, questi iniziano sempre con il carattere ; (puntoevirgola) o # (cancelletto).

5.3 Interpretiamo un file ”INI”

In questa sezione interpreteremo il significato delle varie variabili nel file ”stepper_mm.ini”.

5.3.1 Sezione [EMC]

VERSION = $Revision: 1.9.4.3 $

- Questa variabile setta la versione del file ”INI”

MACHINE = EMC-HAL-STEP-MM

- Questa variabile setta il nome della macchina alla quale in file è destinato

NML_FILE = emc.nml

- Setta il file ”NML” da usare

DEBUG = 0

- Setta il livello di *DEBUG*; zero non stampa nessun messaggio.

5.3.2 Sezione [DISPLAY]

DISPLAY = tkemc

- Setta la GUI (Graphical User Interface) da utilizzare, attualmente ve ne sono parecchie tra le quali:
  - tkemc
  - axis
  - mini
  - usermot

CYCLE_TIME = 0.200

- Intervallo di tempo in secondi tra un refresh e l’altro della GUI

HELP_FILE = tkemc.txt

- Setta il file di aiuto per la GUI

POSITION_OFFSET = RELATIVE

- Setta il modo di visualizzazione delle quote all'avvio del software; Ha due valori ammissibili:
  - RELATIVE: relative all'origine in uso
  - MACHINE: relative allo zero macchina

POSITION_FEEDBACK = ACTUAL

- Setta il modo di visualizzazione delle quote all'avvio del software; Ha due valori ammissibili:
  - COMMANDED: visualizza la posizione di arrivo
  - ACTUAL: visulaizza la posizione attuale

MAX_FEED_OVERRIDE = 1.2

- Valore massimo dell'acceleratore; 1.0 per un massimo del 100%

PROGRAM_PREFIX = /usr/share/emc/ncfiles/

- Path completo dove ricercare i file G-Code

INTRO_GRAPHIC = emc2.gif

- File da visualizzare all'avvio del software

INTRO_TIME = 5

- Tempo in secondi della durata della visualizzazione dell'immagine contenuta in INTRO_GRAPHIC

5.3.3 Sezione [TASK]

TASK = minimilltask

- Nome del controllore

CYCLE_TIME = 0.010

- Tempo in secondi del ciclo del controllore

5.3.4 Sezione [RS274NGC]

PARAMETER_FILE = stepper.var

- Setta il file contenente le informazioni specifiche dell'interprete dei comandi

5.3.5 Sezione [EMCMOT]

EMCMOT = motmod

- Nome del controllore di movimento, attualmente esiste solo "motmod" come controllore di movimento

SHMEM_KEY = 111

- `Non ancora redatto`

COMM_TIMEOUT = 1.0

- `Non ancora redatto`

COMM_WAIT = 0.010

- `Non ancora redatto`

BASE_PERIOD = 50000

- Tempo del ciclo base, in nanosecondi; Dal valore di questa variabile dipende tutto il funzionamento del software

SERVO_PERIOD = 1000000

- Tempo del ciclo del pianificatore di movimento, in nanosecondi; Viene automaticamente arrotondato al multiplo più vicino di BASE_PERIOD

TRAJ_PERIOD = 10000000

- Tempo del ciclo del pianificatore di traiettoria, in nanosecondi; Viene automaticamente arrotondato al multiplo più vicino di BASE_PERIOD

5.3.6 Sezione [HAL]

HALFILE = core_stepper.hal

- File da far interpretare dal comando ”halcmd”

HALFILE = standard_pinout.hal

- File da far interpretare dal comando ”halcmd”

5.3.7 Sezione [TRAJ]

AXES = 3

- Numero di assi controllati dal sistema

COORDINATES = X Y Z

- Nome degli assi

HOME = 0 0 0

- Coordinata della posizione di zero macchina per ogni asse

LINEAR_UNITS = mm

- Unità di misura utilizzata per gli assi lineari

ANGULAR_UNITS = degree

- Unità di misura utilizzata per gli assi angolari

CYCLE_TIME = 0.010

- `Non ancora redatto`

DEFAULT_VELOCITY = 0.424

- Velocità impostata all’avvio della macchina per gli spostamenti in manuale, espressa in ”UNITS” al secondo (mm/s, inch/s, gradi/sec, rad/sec, ecc.)

MAX_VELOCITY = 30.48

- Massima velocità consentita dalla macchina, espressa in ”UNITS” al secondo

DEFAULT_ACCELERATION = 300.0

- Accelerazione impostata all’avvio della macchina per gli spostamenti sia di un solo asse che di un interpolazione di più assi, espressa in ”UNITS” al secondo per secondo (mm/s^2)

MAX_ACCELERATION = 500.0

- Massima accelerazione consentita dalla macchina, espressa in ”UNITS” al secondo per secondo (mm/s^2)

5.3.8 Sezione [AXIS_n]

Nel file ”INI” non troverete una sola sezione AXIS_n, ma ne troverete tante quanto il numero di assi che compone la vostra macchina; uno per ogni asse. Per distinguere una sezione dall’ altra è necessario sostituire il carattere ”n” nella dichiarazione della sezione con un numero intero positivo partendo da 0 (zero).

TYPE = LINEAR

- Definisce il tipo di asse, lineare (LINEAR) o angolare (ANGOLAR)

HOME = 0.000

- Posizione alla quale portare l’asse dopo gli zeri macchina

MAX_VELOCITY = 30.48

- Massima velocità consentita dall’asse, espressa in ”UNITS” al secondo

MAX_ACCELERATION = 500.0

- Massima accelerazione consentita dall’asse, espressa in ”UNITS” al secondo per secondo (mm/s^2)

STEPGEN_MAXVEL = 35.00

- Valore utilizzato in `core_stepper.hal` per settare la frequenza massima del'onda generata, solitamente viene settata ad un valore pari al 110-120% di `MAX_VELOCITY`

STEPGEN_MAXACCEL = 520.0

- Valore utilizzato in `core_stepper.hal` per settare l'accelerazionemassima dela frequenza generata, solitamente viene settata ad un valore pari al 110-120% di `MAX_ACCELERATION`

BACKLASH = 0.000

- Valore utilizzato per recuperare piccoli giochi dell'hardware impiegato per la realizzazione dell'asse

INPUT_SCALE = 200

- Numero di impulsi necessari per muovere l'asse di una "UNITS"

OUTPUT_SCALE = 1.000

- Nell'uso di motori passo-passo questa variabile assume **sempre** il valore 1.0; Serve ad indicare il numero di impulsi inviati dall' encoder per ogni "UNITS" di spostamento dell' asse

MIN_LIMIT = -1000.0

- Il valore minimo che l'asse può assumere, in altre parole il limite software minore; se il limite viene superato il controller blocca l'asse. Espresso in "UNITS"

MAX_LIMIT = 1000.0

- Il valore massimo che l'asse può assumere, in altre parole il limite software maggiore; se il limite viene superato il controller blocca l'asse. Espresso in "UNITS"

FERROR = 1.270

- L'errore massimo ammesso nell'inseguimento della posizione; Se la differenza tra la posizione comandata e quella letta è maggiore del valore settato in questa variabile il controller blocca tutti gli assi e li disabilita

MIN_FERROR = 0.254

- L'errore massimo ammesso nell'inseguimento della posizione negli spostamenti a bassissima velocità. Deve sempre risultare MIN_FERROR <= FERROR

HOME_OFFSET = 0.0

- La posizione del sensore di zero macchina

HOME_SEARCH_VEL = 0.0

- Velocità alla quale viene ricercata la posizione di zero, espressa in "UNITS" al secondo, se la macchina non dispone di sensori di zero settare questa variabile a zero

HOME_LATCH_VEL = 0.0

- Velocità alla quale viene letta la posizione di zero, per una maggiore precisione settare questa variabile a valori bassi

HOME_USE_INDEX = NO

- Indica se la macchina dispone di encoder con segnale "index" oppure no

HOME_IGNORE_LIMITS = NO

- Indica se nella ricerca dello zero l'asse può raggiungere i sensori di finecorsa oppure no

5.3.9 Sezione [EMCIO]

EMCIO = io

- Nome del controllore di input-output, solitamente questa variabile viene settata od "io"

CYCLE_TIME = 0.100

- Tempo ciclo in secondi

TOOL_TABLE = stepper.tbl

- File dal quale vengono letti i valori degli utensili attrezzati per la macchina, numero dell'utensile, diametro, lunghezza ecc.

5.4 Come si calcolano i valori corretti da inserire nel file "INI"

Questa è forese la sezione più importante, dopo aver capito a cosa servono le variabili del file "INI" vediamo come calcolare i valori corretti da inserirvi per avere una macchina precisa, affidabile, performante e soprattutto sicura per l'operatore.

Per prima cosa settiamo nella sezione [TRAJ] il numero di assi e il loro nome, in questo documento consideriamo una macchina a tre assi cartesiani chiamati X, Y e Z, pertanto settiamo:

- AXES = 3
- COORDINATES = X Y Z

Successivamente settiamo le unità di misura, gli assi lineari saranno espressi in "mm" mentre quelli angolari in gradi quindi:

- LINEAR_UNITS = mm
- ANGULAR_UNITS = degree

Infine settiamo la velocità massima, l'accelerazione massima e la velocità ed accelerazione per gli spostamenti manuali.

Questi valori variano molto da macchina a macchina, per una macchina con motori ed azionamenti mediamente performanti (tutte le elettroniche basate sulla coppia L297-L298 e motori adatti) possiamo considerare una velocità massima di circa 2000mm/min e un accelerazione di circa 10000mm/(min*s) quindi setteremo le variabili rispettivamente a `2000/60=33.333` e `9000/60=166.666`, per gli spostamenti in manuale inizialmente setteremo la velocità a 2.5mm/s; Questi valori **non** sono critici, pertanto li arrotonderemo

- MAX_VELOCITY = 30.0
- MAX_ACCELERATION = 160.0
- DEFAULT_VELOCITY = 2.5
- DEFAULT_ACCELERATION = 100.0

Ora passiamo alla sezione [AXIS_0].

Avendo settato nella sezione precedente il nome del primo asse "X" in questa sezione configureremo i parametri con i valori ricavati dalla meccanica dell'asse "X".

Non verranno descritte tutte le variabili avendo già spiegto il loro significato ma solo alcune, le più importanti.

Consideriamo un motore passo-passo da 200 passi a giro pilotato a mezzo passo e collegato alla vite per mezzo di una riduzione con rapporto 1/2 e una vite con passo 4mm.

Per calcolare il numero di passi necessari a far muovere di un millimetro l'asse utilizziamo la seguente espressione matematica:

```
numero_di_passi_motore * numero_microstep / rapporto_di_riduzione /
passo_vite
```

quindi avremo

```
200 * 2 / 0.5 / 4 = 200
```

come descritto in precedenza i valori delle variabili "STEPGEN_MAXVEL" e "STEPGEN_MAXACCEL" non sono altro che il 110-120% delle rispettive "MAX_VELOCITY" e "MAX_ACCELERATION" quindi setteremo

- MAX_VELOCITY = 30.0
- MAX_ACCELERATION = 160.0
- STEPGEN_MAXVEL = 33.00
- STEPGEN_MAXACCEL = 176.0
- INPUT_SCALE = 200

Per una macchina con le caratteristiche sopracitate il valore delle variabili "FERROR" e "MIN_FERROR" dovrebbero andare bene settate rispettivamente ad 1.2 e 0.3 circa.

Per le sezioni [AXIS_1] e [AXIS_2] i calcoli da eseguire sono gli stessi.

Passiamo ora alla sessione [EMCMOT] e calcoliamo i valori per settare le variabili "BASE_PERIOD", "SERVO_PERIOD" e "TRAJ_PERIOD"

BASE_PERIOD è espresso in nanosecondi, nell'uso di EMC2 con motori passo-passo comandati tramite i segnali di passo e direzione (step+dir) indica il tempo minimo che un segnale può rimanere in uno stato, in altre parole è la metà dell'inverso della massima frequenza con cui è possibile pilotare un motore.

Per conoscere il valore di BASE_PERIOD si utilizza la seguente espressione matematica:

```
BASE_PERIOD = 1 / ( INPUT_SCALE * MAX_VELOCITY * 2 ) * 10^9
```

che risulterà dai dati precedentemente settati

```
BASE_PERIOD = 1 / ( 200 * 30 * 2 ) * 10^9 = 83333.333
```

quindi per avere, con la macchina ipotizzata, una velocità di movimento massima di 1800mm/min sarà sufficiente settare BASE_PERIOD ad un valore minore di 83333, questo significa che possiamo tranquillamente lasciare il valore della variabile a 50000.

Il valore assuno dalla variabile BASE_PERIOD non deve essere troppo basso, soprattutto de il software gira su PC ormai vecchi, se settiamo valori bassi la CPU non avrà più tempo per

gestire l'accesso al disco rigido, di aggiornare lo schermo, di controllare la pressione dei tasti sulla tastiera, ecc. con la conseguenza che il computer si blocca e sarà necessario riavviarlo premendo il tastino reset. Per darvi un idea un Pentium III 1000Mhz può lavorare con BASE_PERIOD settata a 14-15000, un AMD Athlon 650Mhz lavora bene settando la variabile a 18-20000, un AMD Athlon 64 3200+ lavora bene anche settando a 9-10000 la variabile.

Vi ricordo che per calcolare la velocità massima di spostamento della macchina basta sostituire i valori ricavati dalla meccanica della macchina alla seguente:

La frequenza massima espressa in Hz è data da `max_freq = ( 10^9 / ( 2 * BASE_PERIOD ) )`

La velocità massima espressa in mm/min è data da `max_vel = ( max_freq / INPUT_SCALE ) * 60`

La variabile SERVO_PERIOD indica il tempo tra due sucessivi aggiornamenti del pianificatore di movimento, è espressa in nanosecondi. In quasi tutti i casi questa variabile va bene settata a 1000000.

La variabile TRAJ_PERIOD indica il tempo tra due sucessivi aggiornamenti del pianificatore di traiettoria, è espressa in nanosecondi. In moltissimi casi questa variabile va bene settata a 10000000, tuttavia su computer veloci è meglio settarla alla metà per avere una migliore granularità del movimento.

5.5 Cos' è un componente "HAL"

Un componente "HAL", Hardware Abstraction Layer, è solitamente un modulo real-time del kernel che implementa una funzione matematica o una serie di operazioni ben precise, che vengono poi utilizzate per creare una macchina complessa. Ad esempio "stepgen" è il componente "HAL" che implementa la generazione dei segnali di step e dir che poi vengono inviati alla parallela tramite "hal_parport" che legge i valori in ingresso e setta i livelli di tensione sulla porta.

Esistono poi anche dei comandi "HAL" utilizzati nei file di configurazione per collegare, in modo virtuale, i vari segnali generati dai componenti.

In definitiva l'Hardware Abstraction Layer è una specie di breadboard software.

5.5.1 Principali comandi "HAL"

In questa sezione verranno illustrati i principali comandi "HAL" e il loro significato.

- `loadrt`: carica un componente "HAL"
- `addf`: inserisce una funzione di un componente "HAL" in un thread per essere eseguita a cadenza certa
- `newsig`: crea un nuovo segnale

- `setp`: setta un pin ad un dato valore numerico
- `linksp`: collega un seganle ad un pin
- `linkpp`: collega due pin tra loro
- `linkps`: collega un pin ad un seganle

5.5.2 Principali componenti ”HAL”

In questa sezione verranno illustrati alcuni componenti ”HAL”.

Parport

Il primo componente è ”hal_parport” che implementa tutte le funzioni necessarie a leggere e scrivere sulla porta parallela del PC.

Questo componente rende disponibili due funzioni denominate ”parport.X.read” e ”parport.X.write” dove la ”X” è un numero intero positivo incluso lo 0 (zero) in base al numero di porte fisiche presenti sul PC. Queste due funzioni vanno inserite in un thread real-time, per mezzo del comando ”addf”, per poter leggere e scrivere i dati a cadenza prefissata.

Oltre alle due funzioni citate ”hal_parport” rende disponibili anche una serie di pin denominati ”parport.X.pin-N-out”, ”parport.X.pin-N-in” e ”parport.X.pin-N-in-not” (dove ”X” e ”N” sono dei numeri interi positivi incluso lo 0) che servono per scrivere o leggere il livello di tensione sul dispositivo fisico.

Stepgen

Questo complnente serve alla generazione dei segnali di passo e direzione in base alla posizione comandata.

Rende disponibili le funzioni ”stepgen.make-pulses”, ”stepgen.capture-position” e ”stepgen.update-freq”

Inoltre rende disponibili i pin ”stepgen.N.counts”, ”stepgen.position-fb”, ”stepgen.N.enable”, ”stepgen.N.position-cmd”, ”stepgen.N.step”, ”stepgen.N.dir”

Ha anche dei parametri: ”stepgen.N.frequency”, ”stepgen.N.maxaccel”, ”stepgen.N.maxvel”, ”stepgen.N.position-scale”, ”stepgen.N.rawcounts”, ”stepgen.N.steplen”, ”stepgen.N.stepspace”, ”stepgen.N.dirsetup”e ”stepgen.N.dirhold”

Per una più approfondita analisi del componente si rimanda all’URL: `http://www.linuxcnc.org/docs/2.1/html/man/man9/stepgen.9.html`

Per saperne di più sui componenti ”HAL”

Per la lista ed i manuali di tutti i componenti ”HAL” si rimanda all’URL: `http://www.linuxcnc.org/docs/2.1/html/` alla ”Section 9” dei ”Manual Pages”.

5.5.3 Come si configurano i pin della porta parallela

In questa sezione verrà preso in esame il file ”standard_pinout.hal”, vedremo come configurare in modo corretto i segnali di passo e direzione ed il segnale necessario a comandare l’accensione dell’elettromandrino.

Per prima cosa notiamo che all’inizio del file vi è il comando

- `loadrt hal_parport cfg=”0x0378”`

questo fa in modo che venga caricato il modulo del kernel richiesto e che venga settato come indirizzo della parallela ”0x378”, se il vostro PC ha due porte parallele o la porta ha indirizzo hardware diverso dovrete correggere questo valore.

Scorrendo il file si incontra il blocco:

- `# next connect the parport functions to threads`
- `# read inputs first`
- `addf parport.0.read base-thread 1`
- `# write outputs last`
- `addf parport.0.write base-thread -1`

Che, come descritto nei commenti, non fa altro che collegare le due funzioni di lettura e scrittura al thread ”base-thread”, si noti che prima vengoo letti i valori dalla parallela e poi scritti, questo per essere sicuri di non muovere gli assi se ad esempio vi è la segnalazione della pressione del fungo di emergenza o se si è raggiunto un limite hardware segnalato dai finecorsa.

Il blocco successivo è quello che descrive i segnali da inviare all’elettronica:

- `# finally connect physical pins to the signals`
- `linksp Xstep => parport.0.pin-03-out`
- `linksp Xdir => parport.0.pin-02-out`
- `linksp Ystep => parport.0.pin-05-out`
- `linksp Ydir => parport.0.pin-04-out`
- `linksp Zstep => parport.0.pin-07-out`
- `linksp Zdir => parport.0.pin-06-out`

Come avrete sicuramente capito basterà modificare i valori dei pin nella maniera richiesta dalla vostra interfaccia.

Scorrendo ancora il file si incontrerà (alcune righe sotto):

- `# create a signal for "spindle on"`
- `newsig spindle-on bit`
- `# connect the controller to it`
- `linkps motion.spindle-on => spindle-on`
- `# connect it to a physical pin`
- `linksp spindle-on => parport.0.pin-09-out`

In questo punto non dovrete far altro che settare il valore adeguato al pin per far in modo che l'elettromandrino venga attivato quando necessario.

Credo che ormai abbiate le idee chiare su come si configura EMC2 e che non sia necessario continuare con le spiegazioni...

Vi ricordo che tutta la documentazione (in inglese) disponibile per EMC2 la trovate all' indirizzo http://www.linuxcnc.org/docs/.

Buon lavoro