

MoonPro .net

Progetti area Mapo/MoonPro

Consiste di vari siti web che ruotano intorno al DB principale MoonPro di seguito elencati che si occupano delle varie gestioni del processo di

- acquisizione dati
- input utente (tramite tablet)
- gestione utente (da browser)
- visualizzazione con analisi (SITE)
- visualizzazione non presidiata (MON)
- [MoonPro .net](#)
 - [Installazione](#)
 - [MP](#)
 - [MP/IO](#)
 - [MP/SITE](#)
 - [MP/TAB](#)
 - [MP/ADM](#)
 - [MP/MON](#)
 - [MP/MAKE](#)
 - [BASIC VERSION](#)
 - [ADV VERSION](#)
 - [HOWTO e gestione problemi](#)
 - [Procedura per ricaricare i dati da rPI-IOB a MP/IO](#)

Installazione

Per installare ci sono dei prerequisiti server:

- IIS
 - Install da roles server x win2012 / win2016
- MS SqlServer Express
 - Scaricare da Microsoft
- Microsoft Sql Server Management Studio (MSSMS)
 - Scaricare da Microsoft
- Redis
 - <https://github.com/MicrosoftArchive/redis/releases>
- Redis GUI, da elenco <https://redis.io/clients> abbiamo selezionato...
 - <https://redisdesktop.com/>
- ..da verificare...
 - <https://www.npmjs.com/package/redis-commander>
 - <https://fastonosql.com/>
 - <https://github.com/leegould/RedisExplorer/releases>
 - <https://github.com/anandtrex/redis-browser/releases>
 - <https://github.com/cinience/RedisStudio/releases>

- <https://github.com/caoxinyu/RedisClient/releases>

MP

- landing site
- rimando a singoli siti

MP/IO

- Verifica status impianti
- Acquisizione da IOB (INPUT DATA)
- Ottimizzazione input tramite layer REDIS x caching dello stato macchine e delle StateMachine degli Ingressi (HW)

MP/SITE

- sito web principale
- visualizzaione stato macchine
- visualizzazione tempi ciclo
- analisi produttività
- verifica ODL correnti/chiusi/aperti

MP/TAB

- sito tablet
- gestione input da operatore
- apertura/chiusura ODL
- gestione causali fermo

MP/ADM

- applicazione amministrativa
- gestione ODL

MP/MON

- monitoraggio continuo (totem officina)
- ottimizzato per rPI
- permette di vedere in 3 modalità
 - Normale (cambia tra 2 versioni colore "fisse") --> MP/MON
 - Blink pulsante (animazione) --> MP/MON/Home/Blink
 - FISSO (NESSUNA animazione) --> MP/MON/Home/Fix

MP/MAKE

- gestione linee di montaggio/assemblaggio ed in genere task manuali
- implementzione pilota e di riferimento x Colcom (sarà poi da integrare con gestionale)

BASIC VERSION

DA RIVEDERE (tolta elettronica scalvina...)

La versione base dell'applicativo gestisce (SENZA integrazione con sistemi informativi esterni) le informazioni collegate alla gestione di fasi manuali.

In particolare si tratta di

- registrare MANUALMENTE i TASK (= gli ordini di produzione, con set minimo: cod_ordine, cod_articolo, qta_richiesta)
- obbligatorio RICONOSCIMENTO dell'operatore (barcode? QRCode?)
- utilizzare il barcode ove possibile (es fase di registrazione TASKS)
- decodificare l'informazione in modo parametrico (es. ordini iniziano per "ORD*", lunghezze minima 10 char...; articoli iniziano per "ART*"; qta da produrre è una cifra < 999'999) --> vedere ad esempio modalità riconoscimento datamatrix in GMW
- al PRIMO caricamento di un TASK (riconosciuto dal codice UNIVOCO dell'ordine di lavoro NON ancora registrato su DB) --> predisposizione alla registrazione del record
- letto il valore CHIAVE (esterna) --> sono accettati i dati a corredo (codice articolo e quantità da produrre)
- conferma una volta letto il set intero --> CREAZIONE DELLA PROMESSA (PromessaODL)
- Una volta creata la promessa (intestata su un GRUPPO GLOBALE) questa diventa attiva e "accettabile" per l'avvio di qualsiasi fase di lavoro (NON E' preimpostato un ciclo, si leva qualsiasi fase venisse associata in futuro)
- Ogni postazione potrà (anche contemporaneamente ad altre) INIZIARE UNA REGISTRAZIONE, ovvero registrare un PERIODO di attività (=fase di lavoro specifica) da associare al TASK, tendenzialmente per una quantità COMPLETA (= tutto quanto indicato come Qta_RICHIESTA)
- Una volta avviata un'attività questa potrà SOLO essere CONCLUSA/SOSPESA. Se SOSPESA si propone DI DEFAULT di riaprirla (es dopo una pausa o il giorno successivo) sullo stesso impianto/postazione. Se viceversa fosse segnalata come conclusa NON verrà proposto di riavviarla (ma non sarà impedito, es fasi di riparazione post collaudo...)
- Possiamo avere TANTE registrazioni di PERIODI di attività per OGNI task attivo (fino a quando il task non viene chiuso). Ogni registrazione DOVREBBE riportare la quantità evasa nel periodo (se non fatto verrebbe sbagliata registrazione tempo ciclo/produttività personale)
- Una volta chiusa l'ultima registrazione, nelle postazioni abilitate si può CHIUDERE il TASK (=lavorazione) e da quel momento NON si può riaprire (tipicamente collaudo, primo collaudo o collaudo successivo per pezzi ripresi)
- Esistono le "fasi standard" (da effettuare su TUTTI i pezzi richiesti) e le fasi di ripresa (tipicamente solo su un subset dei prodotti)
- Output: report base (excel) delle fasi (TUTTE!) associate ad ogni task, con durata TOTALE fase (e quantità pari a quella ordinata - con le fasi di ripresa POTREBBE NON essere vero...)
- Totale tempo fasi / pezzi --> TC stimato
- Si potrebbe costruire un grafico / curva (opzionale, in secondo periodo) per indicare una CURVA del TCiclo al variare della quantità di pezzi lanciati in un ordine di produzione

ADV VERSION

La versione avanzata COMPRENDE tutti i dati, i metodi e le funzioni di base. Si aggiungono però alcune caratteristiche:

- dati ORDINI/FASI recuperati TIPICAMENTE da gestionale / DB+tab esterni
- ...

HOWTO e gestione prolemi

Procedura per ricaricare i dati da rPI-IOB a MP/IO

In caso di problemi (disconnessioni lunghe, PI che non trasmettono) i dati salvati nel logfile dei PI possono essere ritrasmessi al server MP/IO in modalità bath, per farlo:

- mettere la macchina in modalità insEnabled=0 (così quando riparte NON trasmette ma accumula in RAM)
- riavviare il PI
 - collegarsi con WinSCP
 - aprire sessione ssh (ctrl-P x aprire sessione sul PI)
 - diventare root (basta digitare senza pwd il comando **sudo su**)
 - digitare reboot
- quando il PI è ripartito si deve preparare il file... per farlo
 - andare nella directory steamware (/home/pi/steamware) con **cd steamware**
 - copiare il file di log e selezionare il periodo da ritrasmettere, per farlo:
 - **cp logfile.txt oldLog/log_20171124.txt** (conviene salvare con data del log)
 - editare il file con vim (editor di testo) col comando **vim oldLog/log_20171124.txt**
 - cercare ULTIMA trasmissione dati, ovvero quando sul log scrive **offline**, per farlo il comando è **/offline**; questo trova la prima occorrenza, x cercarcel tutte usare **n** che le mostra tutte (in basso indica la % nel file), ci serve ULTIMA VOLTA che va offline (attenzione che se si è riavviato ha scritto offline anche dopo riavvio, non è questo...)
 - eliminare tutte le righe PRIMA dell'ultimo offline (che sono già state salvate sul server...), per farlo quando si rifà la ricerca in absso a dx c'è il numero della riga, numero della colonna e poi % del documento, ci interessa numero della riga...
 - andiamo alla prima riga del file col comando **:1** (invio)
 - scriviamo il comando di cancellazione (**dd**) per il numero di righe trovate (meno qualcuna, nel nostro esempio è 131698--> prendo 131690) e dare il comando **131690dd** (tutto attaccato); se quancosa andasse storto si annulla con il comando **u** (undo)
 - quando lui elimina riporta in absso a sx il risultato (es 131690 fewer lines)
 - se si fa un pasticcio x uscire senza salvare si usa **esc** (anche + volte) e poi **:q!** (invio) per uscire (:q) senza salvare (!)
 - per salvare l'editing comando **w** invio (write)
 - eliminare una x una le singole righe rimaste prima del record offline (si può lasciare) con il comando **dd** (dopo esser certi di essere alla prima riga... **:1**)
 - ora va eliminato la aprte DOPO il riavvio --> vado in fondo (**shift-g**)
 - si risale con PgUp / frecce fino a trovare record riavvio "Avvio Programma ReadPar"
 - si eliminano le righe da li in poi (**1000dd** x eliminarne 100, saranno meno...)
 - si salva e si esce: **:wq**
 - convertiamo il file con il programma **./convertLogFile.py oldLog/log_20171124.txt pMem.dat** (nome programma python, file sorgente, destinazione)
 - verificare il file convertito, DOVREBBE essere nel formato *aaaaMMggHHmmssnnn* (*nnn=millesimo=*)#SEGNALE#contatore

- INVIARE IL FILE!!!! **./readFile.py pMem.dat reload.log** (programma, file_da_inviare file_log_invio)
- RIMETTERE LA MACCHINA ONLINE (e dovrebbe automaticamente caricare i dati che aveva in RAM dal riavvio...)