

Nota generale sull'RTE (Real Time Engine) e sulla visualizzazione delle pagine grafiche nell'ecosistema Syel + Codesys

13-01-2023

Draft, version 1.0

Installazione di RTE sul target P10L

Vediamo come scaricare su un terminale video HMI P10L "vergine" (sia CM3 che CM4, ovvero Raspberry PI Compute Module 3 e 4) il supporto real time, ovvero l'RTE (Real Time Engine) dal PC-host-Windows (x86, x64), dove sopra gira l'ambiente di sviluppo Codesys IDE, poi come scaricare l'applicazione di lavoro da noi sviluppata, e come viene gestita la visualizzazione grafica delle pagine dell'applicazione di lavoro.

Per condurre queste prove, abbiamo collegato via ethernet il nostro PC-host-Windows, con sopra Codesys-IDE (versione 3.5, service pack 17, nel nostro caso), con la porta ethernet del P10L-CM4 (con P10L-CM3 sarebbe la stessa cosa).

Facciamo prima questo:

Ir	n linea 🛛 Debug	Тос	ols	Guida							
	用制制剂	ø	Gestore pacch	netti		Ø → = %	Ç⊒ ⊊⊒ d	= →I 2	\$ + 🎫 🖅 🗸		
		1	Repository lib	rerie							
g	s 📝 🛉 Devic	1	Repository dis	spositivi		EtherCAT_test2	1	TMC460	_select_mode	s4000_DigitalC	Outputs
	Impostazioni co	-	Repository sti	li di visualizzazione							
			Repository de	lle licenze	1						
	Applicazioni	-0	OPC UA Infor	mation Model Repository	- 1						
	Salva e ripristin		Gestore di lice	enze	- 1	i —					
	· · ·	\bigcirc	Lettore del dis	spositivo	Ī						
	File		Personalizza			in the second)		•		
	Log		Opzioni		- 1	Gateway				-	
	209		Importazione	ed esportazione opzioni in corso				\sim	raspberrypi (attivo)		\sim
	Impostazioni PL	•	Scripting		• s	• 42			Nome del dispositivo: raspbenypi		
	Shell PLC		Edge Gateway	(→				Tedalar del dia estima		
			Miscellaneou	s	→	–			0301.3091		
	Utenti e gruppi		Update Edge	Gateway					Target ID:		
	Diritti di accesso		Update Rasph	erry Pi					0000 0011		
	Diritti relativi ai s	simbo	li						Tipo di sistema di destin 4102	azione:	
	Licensed Softwa	ire Me	trics						Fornitore del sistema di 3S - Smart Software Solu	destinazione: ıtions GmbH	
	Elenco attività								Versione del sistema di 0 4.4.0.0	destinazione:	



Codesys on Syel \rightarrow RTE + TargetVISU



Uno schema funzionale di massima:



Da notare che il processo Codesys-Gateway-SysTray deve essere avviato (Start Gateway), se si vuole che il blocchetto Gateway abbia il pallino di color verde e non nero. Con Gateway avviato (pallino verde), Codesys-IDE può, via OPC-UA/TCP, comunicare con P10L (PLC). All'inizio, ovvero con P10L vergine, è normale avere il blocchetto Raspberry PI con pallino nero, questo perché adesso, su P10L (= Raspberry PI by Syel), non abbiamo ancora installato il processo RTE.

Fare quanto segue:



Text Text	NopenExam	ple.project -	CODESYS									
File	Modifica	Visualizza	Progetto	Compila	zione	e	In lin	ea D	ebug	Tools	Finestre	Guida
1 🖻		ഗവ %	ħ®×	M 🖓	6	<u>_</u>	Щ	11 1	[2]	la 🏪	· 🖻 🏙	Applicati
_												
Raspber	rry Pi			•	џ	×	gs	/1	Devic	e x 🔮	HTTPClie	nt_Task
🖌 Logi	n credentia	als					In	nposta	zioni co	municazior	ni	Sfoglia la re
Userna	me	pi					-					1
Passwo	ord		234]	A	pplicazi	oni			
SSH	login based	l on key					Sa	alva e ri	pristina	9		
							Fi	ile				
∡ Sele	ct target						Lo	og				
IP Addr	ess	192.168.25.	107	Scan			In	nposta	zioni PL	с		
⊿ COD	ESYS Runt	ime Package	e ———			1	sł	hell PLC				
	~	3.5.11.20 3.5.16.20 4.0.1.0 (rasp	berry, armhf)				U	tenti e j	gruppi			
Versior	~		oen yy anniny				D	iritti di a	accesso			
							D	iritti rel	ativi ai s	simboli		
							Li	icensed	Softwa	re Metrics		
		Instal		Remove			El	lenco at	tività			
Packag	e directory	C:\Users\Enri	coMolinari\COD	ESYS (St	tato				
🔺 Addi	tional Pack	age				C						
	Install	- Sei	nza nome - Bl	occo note	: ai w	rindo	ows					
Syst	em —		ers\Enrico	mato Vi Molina	sualiz ari\		, FSY	S Con	trol	for Ra	soberry	/ PT
	System Inf	fo				200		2 201				
4 Dunt	imo											

Stiamo quindi selezionando, nell'opportuno path del PC-host-Windows, il run-time RTE che vogliamo installare sul target P10L, in questo caso l'RTE più recente, ovvero il 4.4.0.0. L'IP è ovviamente quello del target P10L.

Clicchiamo sul bottone Install:



Alla fine del trasferimento, abbiamo installato sul target P10L questo pacchetto qui:

Decode Sys Control for Raspberry PI 4.4.0.0.package 13/06/2022 10:33 CODESYS Package 35.948 KB



Sul target P10L, se battiamo il comando "top" da shell, dobbiamo vedere che il processo RTE è installato e correttamente in esecuzione, insieme a tutti gli altri processi concorrenti su Linux:



Adesso Codesys-IDE ci fa questa osservazione:

Impostazioni comunicazioni	Sfoglia la rete Gateway - Dispositivo -	
Applicazioni		
Salva e ripristina		
File		•
Log	Gateway	raspherrypi (attivo)
Impostazioni PLC	IP-Address:	Nome del dispositivo:
Shell PLC	Port:	Indirizzo del dispositivo:
Utenti e gruppi	1217	0301.3091
Diritti di accesso		0000 0011
Diritti relativi ai simboli	Install or update Edge Gateway	di sistema di destinazione:
Licensed Software Metrics	We discovered that on your target is no current 'CODESYS Edge	itore del sistema di destinazione: Smart Software Solutions GmbH
Elenco attività	Gateway for Linux' installed. We highly recommend installing the newest version. Do you want to install the newest version of 'CODESYS Edge Gateway for Linux' on this target?	ione del sistema di destinazione: .0
Stato		
Informazione	Sì No	

Noi clicchiamo SI, quindi si avvia l'installazione sul target P10L di questo secondo pacchetto:

CODESYS Edge Gateway for Linux 4.4.0.0.package 13/06/2022 14:20 CODESYS Package 12.836 KB



Sul target P10L, se battiamo il comando "top" da shell, dobbiamo vedere che il processo codesysedge.bin è installato e correttamente in esecuzione, insieme a tutti gli altri processi concorrenti su Linux:



Questa è la situazione al momento:



Adesso se diciamo al client OPC-UA/TCP di Codesys-IDE di sfogliare la rete, lui trova il server OPC-UA/TCP in ascolto sulla porta 1217, poiché il processo RTE appena scaricato sul target P10L è già operativo, ed è lui che implementa tale server TCP, che fra poco funzionerà prima come bootloader, e poi come supporto real-time.



Codesys on Syel \rightarrow RTE + TargetVISU

itivi 👻 🕂 🗙	gs 🔐 Device 🗙 🧐 HTTPClie	nt_Task Gestore libreria de EtherCAT_test2	TMC460_select_mode s4000_DigitalOutputs
CANopenExample	Impostazioni comunicazioni	Sfoglia la rete Gateway + Dispositivo +	
E I Logica PLC	Applicazioni	7	
CNC settings	Salva e ripristina		
CANbus Examples CANopen Master Examples	File		•
CANopen Slave Examples GONC	Log	Galeway	√ raspberrypi (attivo) √
🥔 GVL 🏈 GVLA	Impostazioni PLC	IP-Address: localhost	Nome del dispositivo: raspbenypi
Gestore libreria	Shell PLC	Port: 1217	Indirizzo del dispositivo: 0301.3091
Cyde_PLC_(PRG)	Utenti e gruppi	Selations disperitive	Target ID:
Gcode_Interpreter (PRG)	Diritti elativi ai simboli	Selezionare il percorso di rete al controllo	
	Licensed Software Metrics	Action of the second	Nome del dispositivo:
POU_CNC (PRG)	Elenco attività		Driver:
	Stato		IP-Address:
POU_RX_Puzzle (PRG)	Informazione		Port:
POU_TCP_Local_Tx (PRG)			1217

Cliccare qui:



Per trasformare questo pallino nero:



in verde.

Adesso Codesys ci dice questo:





Clicchiamo su SI.

Poi facciamo così:

	Gateway	
	ateway-1 🗸 🗸	pberrypi
IF Id	-Address: Pre calhost	mere ESC per interrompere la scansio
Pr 12	ort: 217	
Aggiungi utente del disp	ositivo	×
		~
Nome	pi	
Nome Gruppo predefinito	pi Administrator	
Nome Gruppo predefinito	pi Administrator	
Nome Gruppo predefinito Password	pi Administrator ••••• 1234	~ ~
Nome Gruppo predefinito Password Conferma password	pi Administrator ••••• 1234 •••••	~ ~
Nome Gruppo predefinito Password Conferma password Sicurezza della passwor	pi Administrator ••••• 1234 ••••• d Debole	~
Nome Gruppo predefinito Password Conferma password Sicurezza della passwor	pi Administrator ••••• 1234 ••••• d Debole ✓ La password può essere modificata dall'utento	~ ~
Nome Gruppo predefinito Password Conferma password Sicurezza della passwor	pi Administrator ••••• 1234 ••••• d Debole ✓ La password può essere modificata dall'utente ☐ La password deve essere modificata al primo	e log-in

Riproviamo a collegarci con il target P10L, ma ci viene detto questo, e noi reinseriamo i dati, gli stessi dati appena inseriti:



G	Gateway	
Gatteway-1	\sim	raspberrypi \sim
IP-Address: localhost		Premere ESC per interrompere la scansione.
Port:		
Accesso utente al disposit	tivo	×
Al momento mano dispositivo. Imme che disponga dei d	a l'autorizzazione per eseguir: ettere il nome utente e la pass diritti di accesso necessarie.	e questa azione sul word di un account utente
Nome dispositivo		
Indirizzo dispositivo	0301.306B	
Nome utente	pi	
Password	••••	•
Operazione: Oggetto:	Vista "Device" 1234	
		OK Annulla

Adesso ci viene ripresentato un popup leggermente diverso, e noi inseriamo gli stessi dati:

G	ateway	
Satewayst	✓ ras	spberrypi 🗸
IP-Address: localhost	Pre	emere ESC per interrompere la scansione.
Ports		
Accesso utente al disposit	ivo	×
Al momento manca dispositivo. Imme che disponga dei d	a l'autorizzazione per eseguire que ttere il nome utente e la password liritti di accesso necessarie.	sta azione sul di un account utente
Nome dispositivo	Device (CODESYS Control for Raspb	perry Pi MC SL)
Indirizzo dispositivo		
Nome utente	pi	
Password	••••	٥
Operazione: Oggetto:	Vista "Device" 1234	
		OK Annulla



Adesso il target è pronto, nel senso che non solo il supporto a run-time RTE gira correttamente sul target P10L, ma il nostro PC-host-Windows, tramite il client OPC-UA/TCP di Codesys-IDE, è collegato correttamente al server OPC-UA/TCP implementato dall'RTE. PC e target si vedono, sono connessi, pallino verde.

	Gateway		
Gateway-1	~	raspberrypi (attivo)	~
IP-Address: localhost		Nome del dispositivo: raspbenypi	
Port: 1217		Indirizzo del dispositivo: 0301.306B	
		Target ID: 0000 0011	

Adesso possiamo scaricare l'applicazione di lavoro sul target, facendo il Login:

			Л	Ļ				
izzazi	ione		\sim					
evice	: Logica	PLC]	- Oş	Сğ.)÷	н.	*	C
Q.	68		10 L					
VLA	8	as	se 1 [D	Login	(AL	F+ F8) App	li
enco	elementi							

Poi facciamo il reset a caldo, per portare il program-counter alla prima istruzione dell'applicazione di lavoro:





E adesso clicchiamo su run application:



Adesso l'applicazione gira.

RTE è un framework, una engine, un motore sotterraneo che ha prima funzionato come bootloader TCP in ascolto su una porta, grazie al quale siamo riusciti ad inviare al target l'applicazione di lavoro, e da ora in poi RTE funziona appunto come framework, che fa il parsing di uno script, dove lo script è proprio l'applicazione di lavoro appena inviata.

RTE è un po' come .NET su PC-Windows, l'applicazione di lavoro è un po' come un programma scritto in C-Sharp.

Visualizzazione delle pagine grafiche sul target P10L

Guardiamo qui:

- ₽ X	VISU_TASK TomCat460	u200_2_pot2_display
Task_TCP_Local_Rx (IEC-Tasks) POU_TCP_Local_Rx	Visualizzazione di avvio	Base
Task_TCP_Local_Tx (IEC-Tasks)	Nome del file htm	webvisu
POU_TCP_Local_Tx		🔽 Utilizza come pagina predefinita
POU_TX_PC	Frequenza di aggiornamento (ms)	200
Task_TX_Puzzle (IEC-Tasks)	Dimensione standard del buffer di comunicazione	50000
Task_UDP (IEC-Tasks)	Dimensione ottimale	Visualizza visualizzazioni utilizzate
POU_UDP	Fisso Isotropo	○ Anisotropo
VISU_TASK (IEC-Tasks)	Utilizza le opzioni di messa in scal	a per le finestre di dialogo
VisuElems.Visu_Prg	Larghezza client	1024
VisualizationManager	Altezza client	600
Base		
CNC1_test	Opzioni di rappresentazione	
CNC2_test	Disegna "antialiasizzato"	
CNC3_test	Immissione di testo standard	
CNC4_test	Ingresso a mezzo di	Touchscreen ~
CNC5_test		
	Handle touch as mouse events	



L'applicazione ad alto livello, da noi sviluppata, mette a disposizione un file "webvisu.htm".

Dobbiamo quindi far sì che Linux, su target P10L, lanci un qualunque internet-browser, ad esempio Chromium-Browser, e far sì che tale internet-browser apra (richieda), in local-host (127.0.0.1), la risorsa "webvisu.htm".

In sostanza:

su Chromium-Browser su P10L possiamo battere, sulla barra dell'URL:

http://127.0.0.1:8080/webvisu.htm

/webvisu.htm = end-point, risorsa di cui il browser fa la GET:

GET /webvisu.htm HTTP/1.1 Host: 127.0.0.1:8080 From: Chromium-Browser

Durante lo sviluppo dell'applicazione di lavoro, si può pensare di procedere in questo modo:

mettiamo, in un qualunque path del filesystem di P10L, uno script:



Appena avviato il P10L, lanciamo una shell (di default, sia in "/home/pi", poiché noi siamo utente "pi"), dopodichè battiamo il comando:



sh script2

questo è il contenuto di script2:

chromium-browser --noerrdialogs --allow-running-insecure-content --window-position=0,0 --start-maximixed --window-size=1024,600 --kiosk --incognito http://localhost:8080/webvisu.htm

Pertanto viene lanciato l'internet-browser Chromium (il quale si apre e ricopre tutto l'LCD, 1024 x 600 pixels, in modalità kiosk, ovvero full-screen e foreground, ciò la finestra di Chromium viene avanti a tutto il resto, e qualunque altra pagina/finestra/popup grafici resta sotto), passandogli tutta una serie di parametri, e in particolare gli viene passato il parametro URL:

http://localhost:8080/webvisu.htm

Chromium è quindi un http-client che si connette con l'http-server "localhost" (lo stesso P10L), server che ascolta sulla porta 8080, server implementato dall'applicazione di lavoro da noi sviluppata, o meglio, dal task "VISU_TASK" (libreria "TargetVisu"):

- ┦ X	VISU_TASK TomCat460	u200_2_pot2_display
Task_TCP_Local_Rx (IEC-Tasks) POU_TCP_Local_Rx Task_TCP_Local_Tx (IEC-Tasks)	Visualizzazione di avvio Nome del file htm	Base
POU_TCP_Local_Tx Task_TX_PC (IEC-Tasks) POU_TX_PC Task_TX_Puzzle (IEC-Tasks) Task_TX_Puzzle (IEC-Tasks)	Frequenza di aggiornamento (ms) Dimensione standard del buffer di comunicazione	Utilizza come pagina predefinita 200 50000
Task_UDP (IEC-Tasks)	Dimensione ottimale	<u>Visualizza visualizzazioni utilizzate</u>
□ 🖓 POU_UDP2 □ 🕸 VISU_TASK (IEC-Tasks)	 ● Fisso ○ Isotropo ✓ Utilizza le opzioni di messa in scala 	O Anisotropo a per le finestre di dialogo
Visuelems.Visu_Prg	Larghezza client	1024

Server che restituisce, al client Chromium, la risorsa webvisu.htm richiesta.

Il client, a questo punto, fa il parsing del file webvisu.htm, e da questo parsing viene fuori la visualizzazione grafica delle pagine dell'applicazione di lavoro:



Base
Dase
🖷 💾 CNC1_test
CNC2_test
CNC3_test
CNC4_test
CNC5_test
CNC_error_X
CNC_error_Y
CNC_monitor_X
CNC_monitor_Y
CNC_Select
📲 Cyde1_LD
- 🖶 Cyde2_LD
- 🖶 Cyde3_LD
- 🖶 Cyde4_LD
- 🖶 Cyde5_LD
Esempi_cicli_auto

Il risultato è questo:





"Base" è la pagina di intro, in cui dobbiamo mettere un oggetto di tipo "Frame" delle stesse dimensioni 1024 x 600 dello schermo del P10L e delle pagine grafiche, grazie al quale possiamo indicizzare tutte le pagine del programma:

🗈 CNC1_test 🛛 🛃 💌	Proprietà		
	🍸 Filtro 🔹 🔸 Ordina 🔹 🤶 Ordine 👻	Avanzato	
^	Proprietà	Valore	
100 % 🔍 🗸	Nome dell'elemento	GenElemInst_1	
	Tipo di elemento	Frame	
	Tronca		
	Disegna bordo	Nessun bordo	
	Tipo di graduazione	Anisotropo	
	 Visualizzazioni referenziate 		Configura
	 Posizione 		
	X	13	
	Y	-1	
	Larghezza	1024	
	Altezza	600	
one			
	포 Colori		
	■ Rappresentazione		
	😑 Testi		
	Testo		
	Descrizione comandi		
	Proprietàtesto		
	 Movimento assoluto 		
	■ Movimento relativo		
Presenter	🗷 Variabili a colori		
	 Aspetto variabili 		

Visualizzazioni referenziate		Configura
IntroPage	0	
TMC460_select_mode	1	
TMC460_Speed_control	2	
TMC460_GOTO	3	
Miscell 1	4	
SpeedMotor_Monitor	5	
IO_exp	6	
u200	7	
u200_pot1_display	8	
u200_pot2_display	9	
u200_pot3_display	10	
u200_pot4_display	11	
s4000	12	
s4000_DigitalOutputs	13	
s4000_AnalogOutputs	14	
u200s_sel	15	
u200_2	16	
u200_2_pot1_display	17	
u200_2_pot2_display	18	
u200_2_pot3_display	19	
u200_2_pot4_display	20	



L'avvio manuale dello script "script2", da shell, può andar bene in fase di nostro sviluppo, ma quando si consegna il P10L al cliente, questo appena accende il P10L, deve vedere l'avvio automatico di Chromium-Browser, quindi dobbiamo fare in modo che tale script venga interpretato da Linux (o meglio, dal programma "sh"), in modo automatico, all'avvio del P10L.

Ci sono svariati modi, la descrizione dei quali esula dallo scopo di questo documento: ad esempio uno di questi è di scrivere la riga dello script di cui sopra dentro il file ".profile" che si trova in "/home/pi", poiché lo script ".profile" viene interpretato automaticamente appena l'utente pi effettua il login.