



www.asap-firmware.com

Prefazione



Introduzione
Descrizione del CANopen
Communication Profile DS-301
Device profile DS-401
La soluzione ASAP

Titolo: FWS0007A

Protocollo di comunicazione CANopen

Introduzione

Il CAN è una rete di comunicazione dei dati di tipo industriale, quindi un bus di campo.

Esso è stato ideato inizialmente per ridurre il cablaggio all'interno degli autoveicoli, ma ben presto ha trovato impiego anche nel settore industriale, dove attualmente è uno standard specialmente quando servono requisiti di real-time.

In riferimento allo standard ISO-OSI (parte sinistra) il CAN Bus (parte destra) può essere rappresentato come segue :

Applicazione	Applicazione
Presentazione	
Sessione	
Trasporto	
Rete	
Collegamento	LLC (Logical Link Control) MAC (Medium Access Control)
Fisico	PLS (Physical Layer Signalling) PMA (Physical Medium Attachment) MDI (Medium Dependent Interface)

Sono definiti solo i livelli 1 e 2, i livelli dal 3 al 6 non sono presenti e il livello 7 è definibile liberamente.

Inizialmente questo aveva portato diverse aziende a sviluppare il proprio protocollo cercando poi di diffonderlo il più possibile; sono nati così degli standard *de facto* e tra quelli che più si sono affermati troviamo il DeviceNet della AllenBradley, l' SDS della Honeywell ed il SeleCAN della Selectron.

Contemporaneamente in Germania si era formata un'altra organizzazione a sostegno del CAN nell'automazione denominata CiA (CAN in Automation), indipendente dai prodotti delle aziende e con lo scopo iniziale di divulgare la conoscenza del CAN.

In questo gruppo formato inizialmente da aziende di piccola e media importanza, era nato il desiderio di adottare una strategia comune per non dover sottostare ai protocolli proprietari che si stavano diffondendo; si sono attivati dei gruppi di lavoro che hanno definito così un nuovo protocollo denominato CAL (CAN Application Layer).

Sviluppato con l'obbiettivo di soddisfare tutte le aspettative possibili e di poter essere completato con specifiche particolari, il CAL è risultato piuttosto elaborato e dispendioso nell'implementazione, anche se la sua modularità ne facilita la personalizzazione.

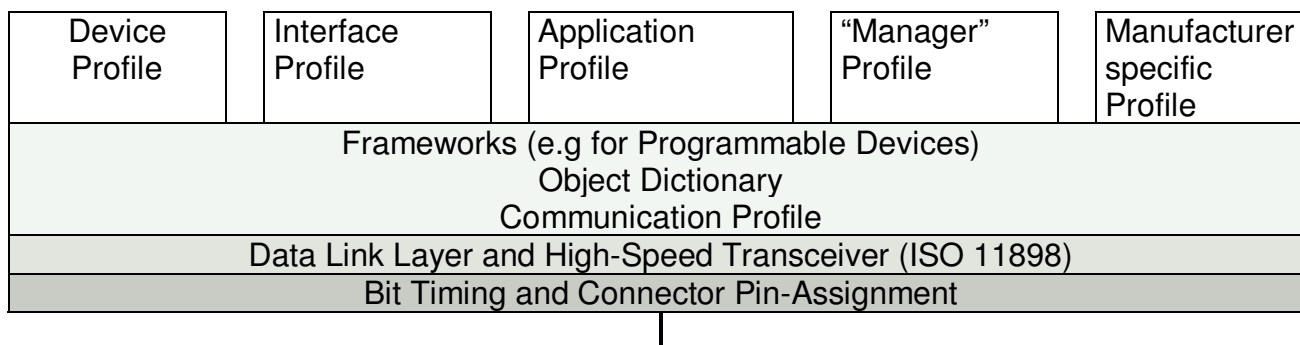
In seguito è nata così l'esigenza di sviluppare un protocollo avente la stessa flessibilità del CAL ma che rispondesse solo ad alcuni sottoinsiemi della norma completa.

Partendo da una versione sperimentale realizzata con il progetto Esprit sotto la presidenza di Bosch, si giunge così al 1995 con l'emissione da parte della CiA della specifica CANopen.

Inizialmente il profilo di comunicazione deriva dal CAL (CiA DS-201...DS-207) e solo nel 1999 viene realizzata la nuova specifica CiA DS-301; oltre a questa troviamo le linee guida per cavi e connettori (CiA 303-1), per dispositivi programmabili (CiA 302) e per unità di misura e prefissi (CiA 303-2).

L'integrazione di funzionalità relative a dispositivi, interfacce e applicazioni specifiche sono facilitate dalle presenza di profili standard (CiA DS-4xx) che interagiscono con i livelli sottostanti relativi alla comunicazione.

Il concetto di comunicazione CANopen viene descritto in conformità al modello di riferimento ISO/OSI come segue :



Bus lines

“Bit Timing” specifica i parametri delle tempistiche di comunicazione :

- Bit rate (da 10kbp/s a 1Mbit/s)
- Durata nominale del singolo bit in funzione del bit rate
- Numero di “quantum” per ogni singolo bit in funzione del bit rate
- Punto di campionamento in funzione del bit rate
- Frequenza di oscillazione (tipica 16MHz +/-0.1%)

“Connector Pin-Assignment” definisce il significato dei pin nei connettori :

- 9-pin D-Sub (DIN 41652)
- Mini Style (ANSI/B93.55M-1981)
- Open Style (morsetto)

“Data Link Layer and High-Speed Transceiver” definisce le specifiche elettriche dei segnali come specificato in ISO 11898 :

- il mezzo di comunicazione è un doppino (tempo di propagazione 5ns/m) con segnali in modo differenziale (V_{CAN_H} e V_{CAN_L}) con ritorno comune; l’isolamento galvanico è opzionale e viene raccomandato per lunghezze del bus superiori ai 200m; è raccomandato anche di utilizzare transceiver in grado di sopportare il contatto accidentale con uno qualsiasi dei segnali sul connettore compresa l’alimentazione

“Communication Profile” definisce i servizi per ricevere e trasmettere gli oggetti sul bus:

- variabili, parametri, numeri o altro vengono chiamati “oggetti” nel mondo del CAN; tra questi troviamo PDOs, SDOs, Special Objects, NMT Objects

“Object Dictionary” rappresenta l’interfaccia verso il software applicativo:

- descrive tutti i tipi di dati, gli oggetti di comunicazione e di applicativo
- definisce il formato dei dati scambiati attraverso la rete e il loro significato, nonché le regole per la codifica e la rappresentazione del loro valore; i bit vengono trasmessi in sequenza di ottetti (byte), dal byte meno significativo al più significativo (stile little endian) ad esempio [b7...b0] [b15...b8] [b23...b16]

“Frameworks” rappresenta un livello aggiuntivo specifico ad esempio per dispositivi programmabili:

- nei dispositivi PLC gli oggetti standard previsti non sono sufficienti per gestire i processi di configurazione e salvataggio delle impostazioni; nella specifica DS301 vers.4.02 è presente una normativa in allegato (Annex A) che era originariamente

specificata in “CiA DSP-302 Framework for programmable CANopen devices” che descrive dei tipi di dati specifici e oggetti di comunicazione aggiuntivi per la verifica dell’integrità dei dati, la loro memorizzazione con un Electronic Data Sheet (EDS), un prompt comandi, multiplexed PDOs ecc.

Tutti i “Profile” descrivono le caratteristiche e le funzionalità opzionali del dispositivo :

- esistono Profile già definiti ad esempio per moduli di I/O (DS-401), misurazione e controllo temperatura e pressione (DS-404), gestione encoder (DS-406), controllo trasmissione idrostatica (DS-408), controllo posizione (DSP-402), controllo porte automatiche (DSP-416), controllo ascensori (DSP-417) ecc.

Descrizione del CANopen

Il concetto centrale di CANopen è basato sull’uso di un “dizionario di oggetti” (Object Dictionary), che raccoglie i dati relativi alla comunicazione ed all’applicazione.

Ogni oggetto nel dizionario è indirizzabile usando un indice (index) a 16 bit e un sotto-indice (sub-index) a 8 bit; questa disposizione è conforme al concetto utilizzato in altri bus di campo (es. MODBUS).

La disposizione del Dizionario Oggetti è la seguente:

Indice (hex)	Oggetto
0000	Riservato
0001-001F	Tipi di dato statici
0020-003F	Tipi di dato complessi
0040-005F	Tipi di dato specifici del produttore
0060-007F	Tipi di dato statici specifici del dispositivo
0080-009F	Tipi di dato complessi specifici del dispositivo
00A0-0FFF	Riservato per usi futuri
1000-1FFF	Area del profilo di comunicazione
2000-5FFF	Area specifica del produttore
6000-9FFF	Area profilo standard del dispositivo
A000-FFFF	Riservato per usi futuri

Tutti i parametri sono contenuti nel Dizionario Oggetti, che nella forma completa è formato da sei colonne come segue:

Index	Object	Name	Type	Attribute	M/O
-------	--------	------	------	-----------	-----

La colonna Index indica la posizione nel dizionario

La colonna Object indica il nome simbolico dell’oggetto (es. DOMAIN, VAR, ARRAY, RECORD)

La colonna Name contiene una descrizione testuale

La colonna Type indica il tipo di dato (es. BOOLEAN, UNSIGNED8, SIGNED16)

La colonna Attribute indica il tipo di accesso visto dal bus verso il dispositivo (es. Read/Write, ReadOnly, WriteOnly)

La colonna M/O significa obbligatorio (Mandatory) o opzionale (Optional)

Per accedere a questi dati sono previsti due meccanismi :

PDO – Process Data Object	SDO – Service Data Object
Canale usato per il trasferimento di dati relativi al processo, quindi in tempo reale	Canale usato per il trasferimento di dati di servizio che non hanno requisiti di tempo stringenti
Messaggi sincroni, asincroni e pilotati da eventi	Messaggi asincroni
Identificatori con alta priorità (basso CAN ID)	Identificatori con bassa priorità (alto CAN ID)
Ottimizzati per scambiare dati in modo efficace e veloce	Ottimizzati per varie applicazioni e trasferimento di grosse quantità di dati ma non critiche nel tempo
Corrispondenza diretta con un oggetto del Dizionario	Accesso ad un oggetto indirettamente tramite indice e sotto-indice
Trasferimento su un unico messaggio	Trasferimento su più messaggi

I messaggi scambiati durante la comunicazione sfruttano un numero limitato di identificatori (CAN Object Identifier, COB-ID), i quali sono definiti per difetto secondo un preciso schema di allocazione :

Oggetto	Identificatore
NMT	0
SYNC	128
TIME STAMP	256
EMERGENCY	129-255
PDO1 (tx)	385-511
PDO1 (rx)	513-639
PDO2 (tx)	641-767
PDO2 (rx)	769-895
PDO3 (tx)	897-1023
PDO3 (rx)	1025-1151
PDO4 (tx)	1153-1279
PDO4 (rx)	1281-1407
SDO (tx)	1409-1535
SDO (rx)	1537-1663
NMT Error Control	1793-1919

Gli oggetti relativi al controllo della rete (NMT, SYNC ecc.) hanno la priorità più alta, seguiti dai PDOs e quindi dagli SDOs; infatti la priorità viene stabilita in “AND wired mode” sui livelli del bus al momento di invio dell’identificatore (CarrierSenseMultipleAccess / CollisionDetection) in cui il livello basso (dominante) prevale rispetto al livello alto (recessivo).

Process Data Object (PDO)

Il trasferimento dei dati in “real-time” è compiuto per mezzo di messaggi denominati PDO, utilizzando uno o due messaggi.

Il tipo di comunicazione può essere associato al modello Produttore/Consumatore: i dati del processo vengono trasmessi da un dispositivo (Produttore) a un altro dispositivo (Consumatore) o a più dispositivi (broadcasting).

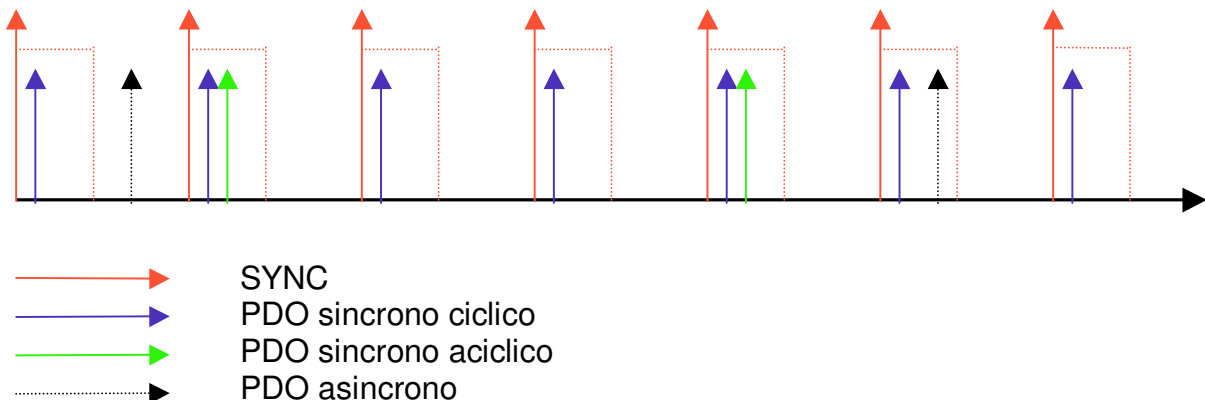
I PDO in trasmissione sono denominati Transmit-PDO (T_PDO), mentre i PDO in ricezione sono denominati Receive-PDO (R_PDO).

Si possono individuare tre tipi di attivazione :

1. evento o tempo
cambiamento di uno stato nel Device Profile, oppure scadenza di un tempo impostabile
2. richiesta remota
ricezione di un RemoteFrame da un altro dispositivo
3. sincronismo
ricezione del messaggio di SYNC

Esiste una ulteriore distinzione in base ai tipi di trasmissione :

4. sincrona
entro un tempo massimo dalla ricezione del messaggio di SYNC (finestra temporale) in modalità :
 - ciclica = legata al numero di SYNC ricevuti
 - aciclica = legata all'evento
5. asincrona
al momento dell'attivazione



Service Data Object (SDO)

Il canale di trasmissione di dati di servizio aventi priorità e criticità del tempo di accesso minori, utilizza i messaggi denominati SDO.

Il tipo di comunicazione può essere associato al modello Client/Server: viene stabilita una comunicazione punto-punto tra il dispositivo proprietario dell'oggetto (Server) e il dispositivo che utilizza l'oggetto (Client).

Questo servizio permette di accedere agli oggetti raccolti nel Dizionario Oggetti, e dato che è possibile definire oggetti a struttura complessa e di notevoli dimensioni (da dati booleani fino a file di programma), questo meccanismo permette un accesso semplice a risorse decentralizzate.

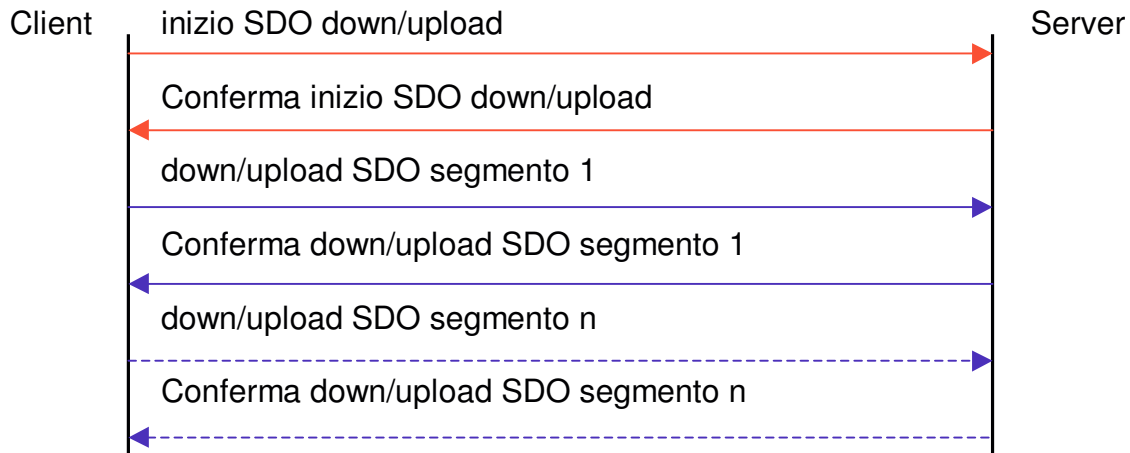
Si possono individuare due modi di operare tramite SDO :

1. trasmissione di dati di dimensione minore o uguale a 4 byte
la trasmissione avviene mediante un pacchetto di invio e uno di risposta e si parla di "protocollo speditivo" (expedited protocol)
2. trasmissione di dati di dimensione superiore a 4 byte
la trasmissione avviene mediante un invio di richiesta di inizio trasmissione alla quale viene risposto, ed una sequenza di invii di segmenti di dato ognuno dei quali viene confermato; si parla in questo caso di "protocollo a segmenti" (segmented protocol)

Il protocollo speditivo è utilizzato per accedere agli oggetti nel Dizionario :

Pacchetto CAN Bus			
Byte0	Byte 1:2	Byte 3	Byte 4:7
Comando	Indice	Sotto-indice	Dati

Il protocollo a segmenti consente ad esempio di trasferire un file eseguibile da un dispositivo ad un altro per eseguire un aggiornamento :



Communication Profile

Nel Dizionario l'area da 1000h a 1FFFh è riservata agli oggetti che contengono i parametri relativi al profilo di comunicazione (Communication Profile).

Indice (hex)	Area
1000-11FF	Tipo dispositivo, registro errori, durata finestra SYNC, versione HW e SW, tempo di heartbeat, COB-ID emergenza ecc.
1200-127F	Parametri SDO Server
1280-13FF	Parametri SDO Client
1400-15FF	Parametri di comunicazione R_PDO
1600-17FF	Parametri di mapping R_PDO
1800-19FF	Parametri di comunicazione T_PDO
1A00-1BFF	Parametri di mapping T_PDO

Nella prima parte della zona sono contenuti gli oggetti che contengono parametri generici come informazioni sul dispositivo, tempistiche e identificatori utilizzati nei messaggi.

Seguono gli oggetti relativi agli SDO lato Server e lato Client, quindi i parametri di comunicazione dei R_PDO (identificatore, tipo di trasmissione, tempi), i parametri di mapping dei R_PDO per associare i dati contenuti ad un oggetto nel Device Profile, e analogamente i parametri di comunicazione e di mapping dei T_PDO.

Il formato di questi oggetti è standard ed è definito dalle relative strutture presenti anch'esse come oggetti (anche se simbolici) agli inizi del Dizionario (area tipi di dato)

Device profile

Nel Dizionario l'area da 6000h a 9FFFh è riservata agli oggetti che contengono i parametri relativi al dispositivo (Device Profile).

In questo caso si è scelto di analizzare un dispositivo di ingresso/uscita generico (generic I/O) sia digitale che analogico, la cui specifica standard corrisponde alla CiA DS-401.

Indice (hex)	Area
6000-61FF	Ingressi digitali
6200-63FF	Uscite digitali
6400-6404	Modulo ingressi analogici
6410-6414	Modulo uscite analogiche
6420-6433	Configurazione ingressi analogici
6440-6451	Configurazione uscite analogiche

Lo scopo dei moduli di I/O è connettere sensori e attuatori nella rete CANopen.

Gli ingressi fino ad un massimo di 2032 vengono trasmessi utilizzando i T_PDO; l'attivazione della trasmissione è causata da un evento come un cambio del loro stato.

Le uscite fino ad un massimo di 2032, vengono ricevute dai moduli preposti e vengono poi trasferite nei relativi driver.

Gli SDO si utilizzano per la configurazione di dati e parametri per la conversione delle misure.

Gli oggetti da inserire dipendono dalle funzionalità che il dispositivo dovrà garantire:

- per gli ingressi digitali è possibile stabilire la polarità, un filtro temporale, la rilevazione del tipo di fronte per segnalare un interrupt
- per le uscite digitali è possibile definire la polarità, lo stato in una condizione di errore, una maschera di attivazione
- per gli ingressi analogici è possibile una lettura con risoluzioni diverse (8,16,32bit float), impostare la condizione per segnalare un interrupt (soglia massima, minima, differenza), un offset e un fondo scala
- per le uscite analogiche è possibile definire lo stato in una condizione di errore, un offset e un fondo scala

La soluzione ASAP

In questa soluzione sfruttando la più che decennale esperienza nella progettazione Embedded e su librerie CAN Bus per protocolli proprietari, si è voluto realizzare una libreria di comunicazione CANopen.

La scelta del Device Profile per I/O è stata fatta perché inerente all'applicazione più comune nel settore dell'automazione, cioè la gestione di I/O remoto; l'approccio modulare seguito nella progettazione consente di utilizzare dei profili di dispositivi diversi senza intaccare lo strato di software relativo alla comunicazione.

Uno degli aspetti più importanti da considerare per l'implementazione del protocollo CANopen su un dispositivo, è la quantità di memoria interna DATA e CODE necessarie.

Sono stati seguiti particolari accorgimenti per ottimizzare l'utilizzo su CPU con risorse interne ridotte (in genere la memoria DATA); ad esempio è possibile definire su quale zona di memoria allocare ogni singolo elemento (sotto-indice) di un oggetto, destinando elementi ad accesso ReadOnly in memoria CODE e mantenendo in memoria DATA solo quelli ad accesso Read/Write.

Uno degli scopi della soluzione consiste anche nel verificare la quantità di memoria richiesta per implementare il minimo dei servizi necessari; questo risulta molto utile in fase di fattibilità di un nuovo prodotto per poter scegliere la CPU adatta e poter ridurre così i costi.

Attualmente sono presenti i servizi di ricezione e trasmissione degli oggetti PDO utilizzando i parametri di comunicazione e di mapping, di gestione SDO in modalità protocollo spedito, di Network Management relativi al BootUp protocol, Heartbeat e Module Control.

A titolo indicativo possiamo riportare che il progetto software compilato con il compilatore Softune Fujitsu per CPU F2MC-16LX riporta per le funzionalità sopracitate un consumo di circa 1900 bytes di memoria CODE e 30 bytes di memoria DATA,

La gestione a basso livello del protocollo (driver CAN Bus) viene gestita in interrupt sia in ricezione che in trasmissione in modo bufferizzato.

Il driver comunica con i livelli superiori attraverso delle funzioni di scrittura e lettura dei messaggi dai due buffer, facilitando la portabilità su piattaforme e sistemi operativi diversi.

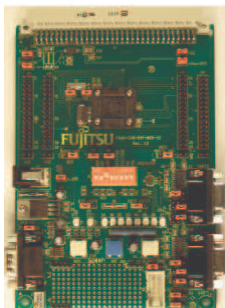
Object Dictionary						
Index (hex)	Sub index	Object	Name	Type	Attribute	Default value
1000		VAR	Device Type	UNSIGNED32	RO	0x00030401
1008		VAR	Manufacturer Device Name	Vis-String4	CONST	"DICE"
1009		VAR	Manufacturer Hardware Version	Vis-String4	CONST	"1.0"
100A		VAR	Manufacturer Software Version	Vis-String4	CONST	"0.90"
1018		RECORD	Identity Object	Identity (23h)		
	1		Vendor – ID (not assigned)	UNSIGNED32	RO	0x00000000
	2		Product Code	UNSIGNED32	RO	0x00000352
	3		Revision Number	UNSIGNED32	RO	0x00000001
1200		RECORD	1 st Server SDO Parameters	SDO Parameters(22h)		
	1		COB-ID Client -> Server	UNSIGNED32	RO	0x00000600 + NodeId
	2		COB-ID Server -> Client	UNSIGNED32	RO	0x00000580 + NodeId
1400		RECORD	RPDO1 communication param.	PDO CommPar.(20h)		
	1		COB-ID	UNSIGNED32	RO	0x00000200 + NodeId
	2		Transmission Type	UNSIGNED8	RO	255
1600		RECORD	RPDO1 mapping parameters	PDO MappPar.(21h)		
	1		1 st mapped object	UNSIGNED32	RO	0x62000108
1800		RECORD	TPDO1 communication param.	PDO CommPar.(20h)		
	1		COD-ID	UNSIGNED32	RO	0x00000280 + NodeId
	2		Trasmission Type	UNSIGNED8	RO	255
	3		Inhibit time	UNSIGNED16	RW	600
	5		Event timer	UNSIGNED16	RW	0
1A00		RECORD	TPDO1 mapping parameters	PDO MappPar.(21h)		
	1		1 st mapped object	UNSIGNED32	RO	0x60000108
6000		ARRAY	8bit digital inputs			
	1			UNSIGNED8	RO	0x00
6200		ARRAY	8 bit digital outputs			
	1			UNSIGNED8	RW	0x00

Node ID= 64 (0x40)

Dal Dizionario si evince la presenza di un RPDO per le uscite digitali e un TPDO per gli ingressi digitali; il valore delle uscite viene trasferito alla ricezione su leds/display, mentre il valore degli ingressi viene spedito ad ogni variazione dovuta al cambiamento del livello sui pins di una porta.

Questa Soluzione è stata realizzata utilizzando come hardware di supporto le schede di valutazione Fujitsu.

Per ulteriori informazioni visitare www.fme.fujitsu.com



La scheda FLASH-CAN-64P-M09-V02 (figura a fianco) ospita il microcontrollore MB90F497:

2Kb RAM, 64Kb Flash, 4MHz x 4 PLL
2 ch UART, controller CAN 2.0B con 8 message buffer
Timers (capture, compare, watchdog)
PPGs
Ext. Interrupt
8 ch AD 8/10bit



La scheda DICE-kit (figura a fianco) ospita il più recente microcontrollore MB90F352 :

4Kb RAM ,128Kb Flash, 4MHz x 6 PLL
2 ch LIN/USART, controller CAN 2.0B con 16 message buffer
Timers (capture, compare, watchdog)
PPGs
Ext. Interrupt
15 ch AD 8/10bit



fa parte dei consulenti riconosciuti da Fujitsu Microelectronics Europe

Per ogni ulteriore informazione, ASAP è raggiungibile al seguente indirizzo di posta elettronica :

info@asap-firmware.com