LABORATORIO: PROVA IN ITINERE

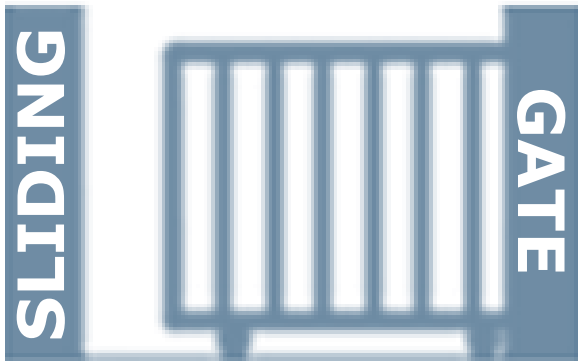
AGE
(Automatic Gate Evolution)

antonino.raucea@dieei.unict.it

Ing. Antonino Raucea

Obiettivo del progetto

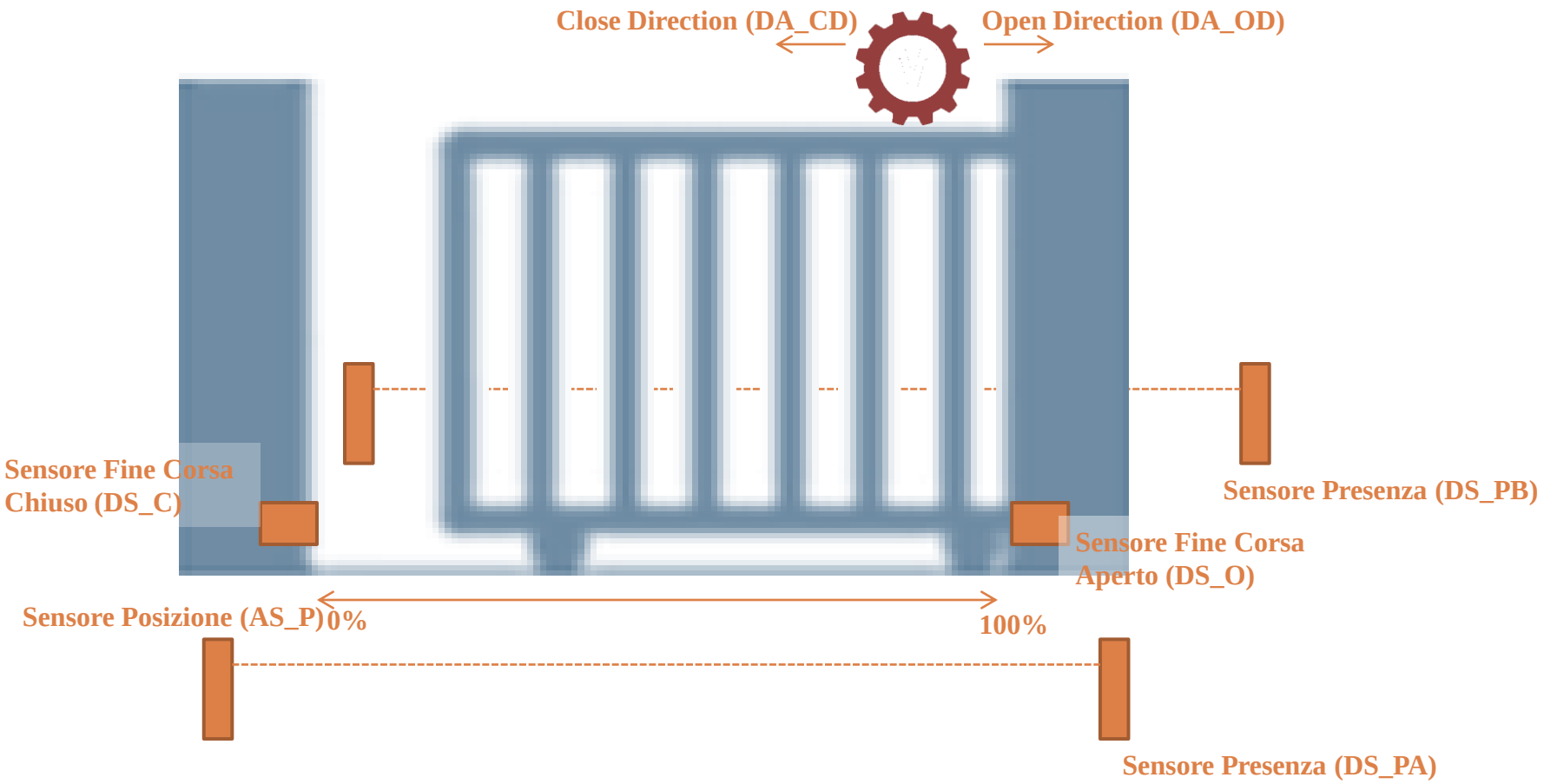
- Control board per Automatic Gate sia di tipo Sliding che di tipo Swing



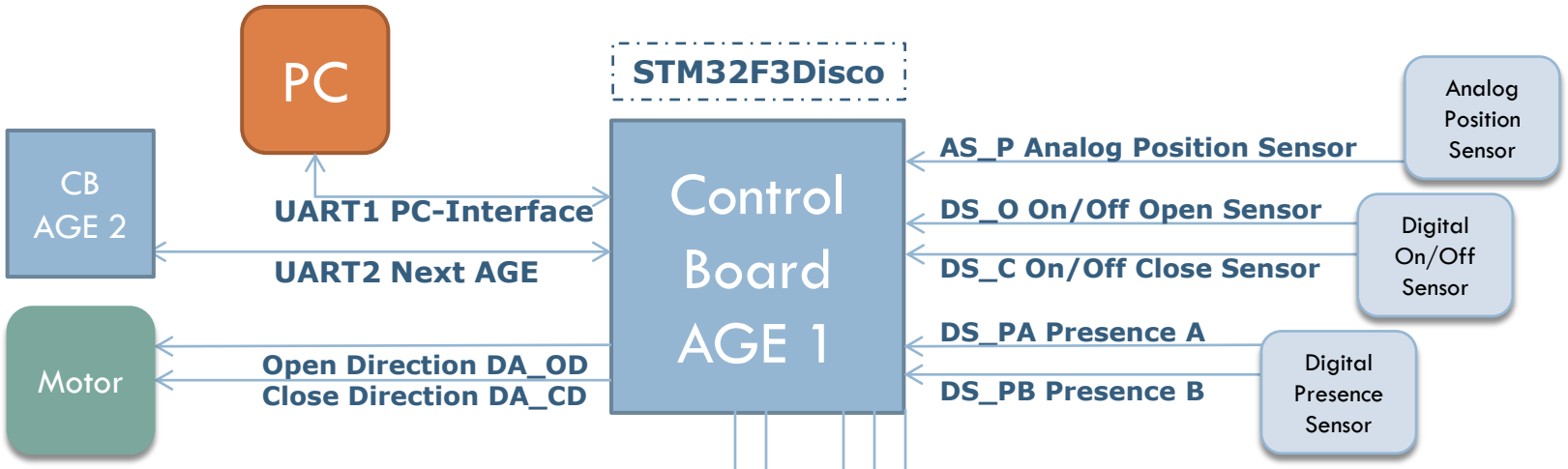
Marketing

- Il progetto prevede la progettazione ed implementazione di un sistema di controllo per cancelli automatici (Automatic Gate) sia scorrevoli (Sliding) che a battente (Swing)
- Il sistema offre funzionalita' evolute quali:
  - programmazione ed controllo attraverso seriale (UART)
  - controllo in cascata di piu' schede bridge
  - sensori presenza, movimento, fine corsa posizione

System Overview



Block Overview



Motor State Indication Led 4

Idle	Off
Motion	On
Error	
TO_O	Blink 0.5 sec
TO_C	Blink 0.2 sec
TO_M	Blink 0.1 sec

Gate State Indication Led 3

Close	On
Opening	Blink 0.4 sec
Open	Off
Closing	Blink 0.2 sec
FreePosition	Blink 1.0 sec

GateState DO_ST

000 Close
001 Opening
010 Open
011 Closing
100 Free Position
1-- Reserved For Future Use

Note

- Il protocollo e' di tipo ASCII
- Tutte le UART sono configurate 115200 8n1
- Se un comando di applicazione (vedi a seguire la definizione) non e' diretto per tale nodo ritrasmetterlo sulla UART2, I pacchetti ricevuti dalla UART2 ritrasmetterli sulla UART1 senza interpretarli (scartare eventuali byte che non appartengono ad un pacchetto).

PINs

Description	Periferica		PIN	AF	DIR	
PC<->ControlBoard	USART1	TX,RX	A9,A10	7	OUT,IN	
Bridge	USART2	TX,RX	A2,A3	7	OUT,IN	
Sensore di posizione	ADC1	ADC2_IN1	A4	-	IN	
Sensore di fine corsa Open	GPIOA	6	A6	-	IN	
Sensore di fine corsa Close	GPIOA	7	A7	-	IN	
Sensore di presenza A	GPIOA (userbutton)	0	A0	-	IN	
Sensore di presenza B	GPIOA	1	A1	-	IN	
Status Bit	GPIOE (Led 5,7,9)	10,11,13	E10,E11,E12	-	OUT	
Led 3 (State feedback)	LED3	-	E9	-	OUT	
Led 4 (Motor status)	LED4	-	E8	-	OUT	
Motor Open Dir	LED6	-	E15	-	OUT	
Motor Close Dir	LED8	-	E14	-	OUT	
Emulator (non si devono configurare)						
Emulator DAC	DAC1	OUT1	A4	-	OUT	Internal connected
Emu Fine Corsa Open	GPIOC (A6)	4	C4	-	OUT	
Emu Fine Corsa Close	GPIOC (A7)	5	C5	-	OUT	

Step1 - Implemetation

- Implemetare un automatic gate board control
- Funzionalita'

a) Basic functionality

b) Open Gate

c) Close Gate

d) Set position

e) UART Protocol:
   - Configuration / Ctrl Command
   - App command
   - UART2 Bridge

Step 1a – Basic functionality

- Predisporre il progetto e creare le strutture per ospitare le seguenti informazioni di configurazione:
  - magic_number int32
  - name : string[8]
  - id : int
  - code : string[4]
  - type : enum Type_e TypeSliding, TypeSwing
  - to_open : int
  - to_close : int
  - to_move : int
- Inizializzazione delle periferiche e dei GPIO
- Setting dello stato e dei relativi feedback attraverso il led, in particolare per il blinking
- Gestione dei sensori ed attuatori sia digitali che analogici

Step1b - Open Gate

- Open Gate
  - Mutare lo stato in @opening 001
  - Configurare l led3 e led4 in accordo con lo stato del sistema
  - Attivare il Pin DA_OD “Open direction” (il pin DA_CD deve essere settato al valore 0) fino a quando il sensore di fine corsa DS_O non restituisce 1
  - Quindi mutare lo stato in @open 011

Step1c – Close Gate

- Close Gate
  - Mutare lo stato in @closing 011
  - Configurare l led3 e led4 in accordo con lo stato del sistema
  - Attivare il Pin DA_CD “Close direction” (il pin DA_OD deve essere settato al valore 0) fino a quando il sensore di fine corsa DS_C non restituisce 1
  - Quindi mutare lo stato in @close 000

Step1d – Set Position

- Set Position
  - Mutare lo stato in @opening 001 o @closing 011
  - Configurare I led3 e led4 in accordo con lo stato del sistema
  - Attivare il Pin DA_OD o DA_CD (l'altro pin DA_xD deve essere settato al valore 0) fino a quando il sensore AS_P non restituisce l'equivalente valore di posizione desiderato
  - Quindi mutare lo stato in @open 010, @close 000, @free_position 100 in accordo con quanto restituito dal sensore AS_P.

Step1e – UART Protocol

- Implementare un protocollo di controllo e gestione attraverso la UART1 secondo quanto specificato a seguire,
- si ripete e si ricorda che se l'id specificato nel pacchetto non coincide con quello configurato nella control board lo stesso pacchetto deve essere reindirizzato verso la UART2,
- Ad ogni cambio di stato si deve inviare una notifica alla UART1 attraverso il pacchetto specifico (vedi protocollo UART)

Step1d - Struttura pacchetto

#<comand>[:<id>[:<code>]][:[<param1>,[…<param N>]]<EOP>

Esempi

#VER\r

#PING:1:1234\0

#OPEN_GATE:2:1234\n

#NOTIFY:2:I,STATE OPEN\n

Dove

Comand	comando valido spiegato nelle tabelle succesive
Code	il codice che identifica in modo univoco il cancello
Param X	i parametri in relazione al comando
EOP	il terminatore del pacchetto che puo' essere uno dei seguenti caratteri '\0', '\r', '\n'

Step1d - Commands

I comandi si suddividono in:

- Comandi di controllo (Ctrl Command)

  sono comandi utilizzati principalmente per configurare la control board e settare name, code, id non hanno bisogno di id e code

- Comandi di applicazione (App Command)

  sono comandi legati all'applicazione e necessitano sempre di id e code, la board che riceve un App Command controlla l'id, se coincide con il proprio allora esegue il comando altrimenti lo ritrasmette sulla UART2 per continuare la catena

Step1d - Ctrl Commands

Get : Request:No parameters
Set : Request: Value[s]

} Response:Value[s] (comma separated)

Command	Id	Code	Access	Note
VER	No	No	R	FW Version (AEG 1.0)
NAME	No	No	R/W	Name (Max length 8)
ID	No	No	R/W	Id (1..9)
CODE	No	No	R/W	Code (Numric)
TO_OPEN (TO_O)	No	No	R/W	Time (sec) 20 – 100
TO_CLOSE (TO_C)	No	No	R/W	Time (sec) 20 – 100
TO_MOVE (TO_M)	No	No	R/W	Time (sec/10) 1 – 50

Step1d - App Commands

Command	Id	Code	Params	Note
OPEN_GATE	Yes	Yes	No param	
CLOSE_GATE	Yes	Yes		
POS_GATE	Yes	Yes	0% - 100%	
STATUS_GATE	Yes	Yes	Request: none Response State,Position	State del sistema open, opening, close, closing, free_position Position rappresenta la posizione attuale del cancello (di fatto l'ultima lettura di AS1 convertita percentuale)
RESET	Yes	Yes	No param	Effettua un reset del sistema
NOTIFY	Yes	No	Tipo,Message	Spedito dalla board al PC per gestire notifica informative e di errore

Step 1d – Notify format

- Type: I = Information, W = Warning, E=Error
- Message
  - State <state> open, opening, close, closing, free_position in accordo con state pin

Step 2 – Sensori di presenza

- Aggiungere due sensori di presenza A e B (o di passaggio)
- I due sensori sono collegati a due pin configurati in input DS_PA e DS_PB, ad ogni loro cambiamento di stato deve essere inviata una notifica attraverso la UART1 #NOTIFY:<id>:W,{P{A|B}_ON|P{A|B}_OFF}\n
- Se il cancello era in apertura, rimane tale, invece se era in chiusura prevedere il cambio in apertura (con aggiornamento dei led e pin di stato).

Step 3 – Error Condition Detection

- Inserire la gestione e notifica delle seguenti condizioni di errore

1. Il tempo per completare un'escursione deve essere minore del TO_OPEN o del TO_CLOSE (in relazione all'operazione in corso), se scatta il TO il motore deve essere bloccato, il led3 lampeggiare in condizione di errore, lo stato aggiornato secondo i valori letti da AS_P ed una notifica inviata sulla UART1
2. Quando il motore e' attivo (in open o close) si deve controllare l'effettivamente movimento (attraverso il sensore AS_P). AS_P non cambia per TO_MOVE mettersi in condizione di errore led 3 ed inviare uno notifica attraverso alla UART1 ed aggiornare lo stato della board con quanto letto