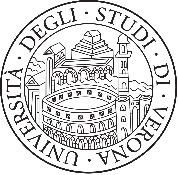
##### UNIVERSITÀ DEGLI STUDI DI VERONA



ELABORATO ASM

Architettura degli Elaboratori Laboratorio

Corso di Laurea in Informatica

A.A. 2022/2023

**PROGETTO:  
Progettazione di un menu per la gestione del menù cruscotto di un’automobile**

**Studenti:**

Alessia **Gennari** VR488137Mattia **Pacchin** VR461870

Sommario

Specifica del progetto 3

Caratteristiche del codice 3

Suddivisione dei file …………………………………………………………………………..3

Variabili 4

Descrizione delle variabili ed il loro scopo 4

Descrizione delle modalità di passaggio delle variabili tra funzioni 4

attuazione 5

Pseudo-codice ad alto livello 5

Scelte progettuali 8

Criticità riscontrate e soluzioni 8

# Specifica del progetto

Il presente elaborato descrive la progettazione di un codice in Assembly AT&T per la gestione di un cruscotto di un’automobile.

## Caratteristiche del Codice

Il menù si può accedere in due modalità: utente e supervisore.

Nella modalità utente è possibile visualizzare il menù composto nel seguente modo:

1. Setting automobile:

2. Data: 15/06/2014

3. Ora: 15:32

4. Blocco automatico porte: ON

5. Back-home: ON

6. Check olio

Nella modalità supervisore (si accede aggiungendo al nome dell’eseguibile il codice 2244) il menù visualizzato sarà il seguente:

1. Setting automobile (supervisor):

2. Data: 15/06/2014

3. Ora: 15:32

4. Blocco automatico porte: ON

5. Back-home: ON

6. Check olio

7. Frecce direzione

8. Reset pressione gomme

In entrambe le modalità sarà possibile spostarsi tra i vari menù con le frecce su e giù).

Premendo la freccia a destra sarà possibile modificare i punti 4, 5, 6, 7 e 8.

* Per i punti 6 e 8 verrà visualizzato un messaggio di conferma.
* I punti 4 e 5 potranno essere modificati premendo freccia in su o freccia in giù, questo modificherà il loro valore da ON a OFF e viceversa.
* Per modificare il punto 7 si richiede di inserire un numero, di default è 3 il massimo è 5 ed il minimo 2, se verrà inserito un numero maggiore di cinque verrà salvato cinque, se invece viene inserito un numero minore di due verrà salvato due.

## Suddivisione dei file

Il codice assembly è diviso in vari file, uno per ogni funzione.

**main.s** : contiene le variabili globali, controlla se è stata attivata la modalità supervisore, chiama una prima volta la funzione index\_position\_message (per visualizzare il primo messaggio) e poi all’interno di un ciclo while richiama la funzione navigate\_menu.

**move.s** : è la funzione che si occupa di prendere in input le frecce e restituire -1 in caso di freccia in su, 1 per freccia in giù e 2 per freccia a destra.

**set\_blinkers.s** : prende in input il numero desiderato di blinks, controlla se è nel range corretto (da 2 a 5) se è maggiore di cinque verrà impostato a cinque se è minore di due verrà impostato a due.

**index\_position\_message.s** : uno switch case che a seconda della posizione dell’index visualizza l’impostazione corretta del menù. Nel caso dei punti modificabili viene effettuato il controllo se è stata premuto il tasto freccia destra, in caso positivo richiama la funzione per poter modificare il valore (es. per modificare i lampeggi richiama la funzione set\_blinkers).

**navigate\_menu.s** : richiama la funzione move, se questa ha dato risultato due controlla che l’index corrente sia modificabile in caso positivo viene messo un flag a true. In caso move non dia come risultato 2 (quindi o 1 o -1) verrà aggiunto il risultato al corrente index tenendo conto se si è oppure no in modalità supervisore.

# VARIABILI

## Descrizione delle variabili ed il loro scopo

**ind** : variabile globale che salva la posizione (index) nel menù. Inizialmente ad 1, viene modificata dalla funzione navigate\_menu .

**sub** : variabile globale che assume valore 1 se è stato premuto il tasto freccia destra per una funzione del menù modificabile, altrimenti vale 0.

**door\_lock**: variabile globale che salva il valore della funzionalità del menù *blocco automatico porte* (4). Di default a 1, che corrisponde ad ON, se viene premuta freccia destra e poi freccia su/giù il valore si modifica e va a 0 (OFF). Se invece il valore salvato è zero allora premendo freccia su/giù la variabile va a uno.

**back\_home** : variabile globale che contiene il valore della funzionalità *Back-Home* del menù. Si modifica con lo stesso funzionamento della variabile door\_lock.

**blinkers** : variabile globale che salva il numero di lampeggi dell’automobile in autostrada (funzione *frecce direzione*). Questa variabile viene modificata dalla funzione set\_blinkers.

Per ogni funzione vengono usate delle altre variabili temporanee che servono per salvare per breve tempo i valori, fare calcoli e poi restituirlo.

## Descrizione delle modalita’ di passaggio delle variabili tra funzioni

// TODO

# attuazione

## pseudo-codice ad alto livello

**# Pseudocodice**

// initiate some global variables shared by all functions

ind = 1 // index var

sub = 0 // submenu var

door\_lock = 1 // bool, true if doors auto lock is on

back\_home = 1 // bool, true if back home mode is on

blinkers = 3 // how many times blinkers blink (supervisor mode)

**### MAIN(terminal input) // return int**

    supervisor = 0

    if terminal input = 2244

        supervisor = 1

    index\_position\_message(supervisor);

    while (true)

        navigate\_menu(supervisor)

    return 0

**### INDEX\_POSITION\_MESSAGE(supervisor)**

    switch (ind)

        case 1:

            if (supervisor = 1)

                print "1. Setting automobile (supervisor):"

            else

                print "1. Setting automobile:"

        case 2:

            print "2. Data: 15/06/2014"

        case 3:

            print "3. Ora: 15:32"

        case 4:

            if (sub)

                read = move()

                if (read = -1 or read = 1)

                    door\_lock = not(door\_lock)

                sub = 0

            if (door\_lock)

                print "4. Blocco automatico porte: ON"

            else

                print "4. Blocco automatico porte: OFF"

        case 5:

            if (sub)

                read = move();

                if (read = -1 or read = 1)

                    back\_home = not(back\_home)

                sub = 0

            if (back\_home)

                print "5. Back-home: ON"

            else

                print "5. Back-home: OFF"

        case 6:

            print "6. Check olio"

        case 7:

            if (sub)

                set\_blinkers()

                sub = 0

            print "7. Frecce direzione: \*blinkers\*"

        case 8:

            if (sub)

                print "Pressione gomme resettata"

                sub = 0

            print "8. Reset pressione gomme"

**### NAVIGATE\_MENU(supervisor)**

    read = move()

    if (pressed right arrow and index has a submenu)

        sub = 1

    else if (pressed up or down)

        ind = ind + read

        if (supervisor = true)

            if (ind < 1)

                ind = 8

            else if (ind > 8)

                ind = 1

        else

            if (ind < 1)

                ind = 6

            else if (ind > 6)

                ind = 1

    index\_position\_message(supervisor);

**### MOVE() // return int**

    get char c

    if (c = up)

        return -1

    else if (c = down)

        return 1

    else if (c = right)

        return 2

    return 0

**### SET\_BLINKERS()**

    get n

    if (n > 5)

        blinkers = 5

    else if (n < 2)

        blinkers = 2

    else

        blinkers = n

# SCELTE PROGETTUALI

//TODO

## criticita’ riscontrate e soluzioni

//TODO