Robot Demo

Obiettivi:

- Programmare Arduino Robot per mettere in evidenza le funzionalità e le diverse risorse hardware presenti.

Risorse hardware & software utilizzate:

- Arduino Robot
- Software per la programmazione: Arduino IDE

Criteri di progetto:

Per meglio comprendere i passaggi necessari alla creazione del dispositivo descritto nei punti sopra illustrati si dividerà la parte esplicativa in due sezioni: hardware e software.

Hardware

Il robot si presenta con una serie di componenti già integrati nel robot, come, ad esempio:

- Bussola
- Potenziometro
- Schermo LCD
- Altoparlante
- Lettore schede microSD (formattate in FAT16 per compatibilità)
- Key pad composto da cinque pulsanti
- Cinque sensori IR (=infrarosso)

Grazie alla notevole integrazione il programmatore non dovrà effettuare modifiche o aggiunte per lo sviluppo di questo progetto.
É molto importante comprendere la distinzione fra *control board* e *motor board*.
La prima (="scheda di controllo") è quella che si programma usualmente e sulla cui superficie sono saldati la maggior parte di sensori e trasduttori; inoltre possiede 8 connettori ciascuno dei quali composto da 3 pin: GND, dato, +5V.
Lo schermo LCD estraibile da 128 x 160 pixel, facilmente individuabile nella parte centrale del robot, ha sul retro l'alloggiamento per la scheda microSD. Quest'ultima ha la funzione di contenere, ad esempio, file

musicali (in formato *.sqm*) o immagini (in .bmp) che mediante le funzioni dedicate potranno essere riprodotti da Arduino Robot.

La motor board (="scheda motori") è responsabile della gestione dei motori e di alcuni altri sensori. Su essa è presente l'alloggiamento per 4 batterie NiMh ricaricabili da 1,5 V ciascuna. Inoltre si possono individuare 4 connettori simili a quelli della *control board*, ma con la differenza che il cui pin dato è analogico. Sulla parte inferiore della *motor board*, che potenzialmente sarebbe a contatto con il suolo, è applicata una maschera protettiva in cartoncino che evita possibili danni alle piste sulla basetta. Un danno alle piste potrebbe compromettere il collegamento tra componenti e quindi causarne un malfunzionamento se non addirittura il guasto.

Software

Il software è stato scritto per mostrare due diversi gradi di autonomia del robot.
Il primo grado mette in evidenza che il robot, gestito dal microcontrollore, esegue le istruzioni scritte nel file sorgente, il secondo grado di *pseudo*-indipendenza unisce le istruzioni previste dal programmatore con delle variabili (velocità e tempo) che in quel momento saranno *pseudo-casuali*. Il risultato che ne deriva sarà ogni volta differente.

```
#include <ArduinoRobot.h> // necessaria per utilizzare il robot
#include <Wire.h> // a seconda della versione di Arduino IDE è opzionale
#include <SPI.h>  // a seconda della versione di Arduino IDE è opzionale
#include "Def_funzioni.h" // libreria creata per la definizione di alcune funzioni


void setup() {
  Robot.begin();  // inizializza il robot
  Robot.beginTFT(); // inizializza lo schermo
  Robot.beginSD(); // inizializza l'SD
  Robot.beginSpeaker(); // inizializza lo speaker

  Robot.stroke(0, 0, 0);  // imposto colore nero per il testo
  Robot.textSize(4);  // imposto dimensione carattere a 4 punti
  Robot.text("CIAO!", 10, 70);  // stampa il testo nel punto P = (10, 70)

  delay(3000);  // ritardo per consentire di spiegare
  Dipendente();   // chiamata f.ne che permette al robot di andare avanti, dx, sx, indietro

  delay(5000);  // ritardo per consentire di spiegare
  Indipendente(); // permette al robot di muoversi in direzione e con tempi pseudo-casuali
  delay(1000); // piccola pausa

  Robot.stroke(0, 0, 0);  // imposto colore nero per il testo
  Robot.textSize(1);  // imposto dimensione carattere a 1 punto
  Robot.text("Grazie per l'attenzione!", 0, 80);  // stampa il testo nel punto P = (0, 80)
}

void loop() {
  // nulla da ripetere
}
```

Lista funzioni Arduino e libreria "Robot" utilizzate:

- Robot.begin();
  Serve ad inizializzare il robot. Nessun parametro e nessun ritorno.
- Robot.beginTFT(*foreGround, backGround*);
  Si utilizza per inizializzare lo schermo. Due parametri: colore in primo piano e colore su sfondo. Nessun ritorno. Se non si assegnano valori ai parametri, lo schermo LCD si imposta da default.
- Robot.beginSD();
  È necessaria per inizializzare la lettura della SD. Nessun parametro e nessun ritorno.

- Robot.beginSpeaker();
  Serve per inizializzare lo speaker montato sulla *control board* di Arduino Robot. Nessun parametro e nessun ritorno.

- Robot.stroke(*red, green, blue*);
  Imposta il colore che si utilizzerà per scrivere, disegnare oggetti. I parametri sono 3 e corrispondono alle componenti RGB. Il valore accettato per ciascun parametro può variare tra 0 e 255. Nessun ritorno.

- Robot.textSize(*size*);
  Imposta la dimensione del testo che si andrà a "stampare" su schermo. È previsto un solo parametro: dimensione del testo espressa in pixel. Nessun ritorno.

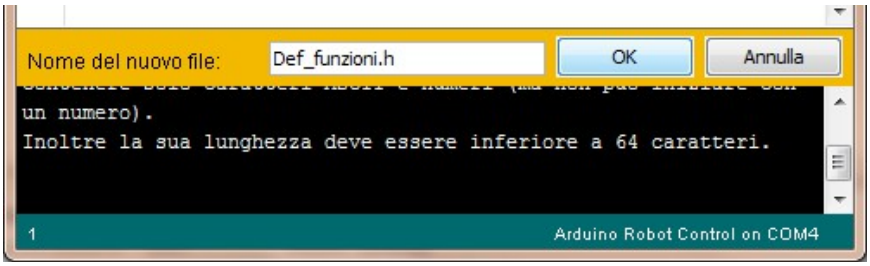
- Robot.text(*toWrite, x, y, writeOrErase*);
  Stampa a schermo ciò che è contenuto nel primo argomento. I parametri sono 4: il primo deve contenere il testo che può essere di tipo *string*, *int* o *long*; il secondo e il terzo contengono rispettivamente l'ascissa e l'ordinata da cui far partire la scritta. L'ultimo parametro se assume valore TRUE per scrivere e FALSE per cancellare. Nessun ritorno.

- delay(*ms*);
  Sospende per il tempo indicato dal parametro l'esecuzione del programma. *ms* è espresso in millisecondi ed è dichiarato come *unsigned long*. Nessun ritorno

Come si può vedere nel codice sopra, compaiono due funzioni *Dipendente()* e *Indipendente()*, esse sono definite in una libreria creata appositamente per questo *sketch*.
La libreria è chiamata dalla direttiva per il precompilatore con la seguente sintassi: *#include "Def_funzioni.h"*. Per creare una nuova libreria si dovranno seguire i seguenti tre passaggi:

1. Click su freccia in basso nell'angolo destro dell'Arduino IDE

2. Nominare la libreria e far seguire il nome obbligatoriamente dall'estensione *.h* .
   Premere su "OK".



3. Se tutto è stato fatto correttamente dovrebbe comparire una seconda scheda nell'ambiente di sviluppo.

Contenuto dell'*header* "*Def_funzioni.h*".

```
void Dipendente()
```

```
{
  Robot.motorsWrite(150, 150);  // robot va avanti
  delay(1000);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(-100, -100);  //robot torna indietro
  delay(1500);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);

  Robot.motorsWrite(100, -100); //robot gira
  delay(700);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(150, 150);  // robot va avanti
  delay(1000);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(-100, -100);  //robot torna indietro
  delay(1500);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(-100, 100); //robot si rigira
  delay(700);
  Robot.motorsStop();           //ferma motori istantaneamente

  Robot.motorsWrite(-100, 100); //robot gira dall'altra parte
  delay(700);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(150, 150);  // robot va avanti
  delay(1000);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(-100, -100);  //robot torna indietro
  delay(1500);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(100, -100); //robot si rigira
  delay(700);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);

  Robot.motorsWrite(-150, -150);  // robot va indietro
  delay(1000);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);
  Robot.motorsWrite(100, 100);  //robot va avanti
  delay(1500);
  Robot.motorsStop();           //ferma motori istantaneamente
  delay(500);

}

void Indipendente()
{
  Robot.clearScreen();  // cancella tutto dallo schermo
  Robot.drawBMP("Giulio_1.bmp",0,0);  // stampa su schermo la foto contenuta nell'SD
  delay(400);
  Robot.playFile("melody.sqm"); // riproduce file contenuto nell'SD

  for(int i = 0; i < 6; i++)
  {
    Robot.motorsWrite(random(-200,201), random(-200,201));
    delay(600);
    Robot.motorsStop();
    delay(500);
    Robot.motorsWrite(100, 100);
    delay(random(750,1300));
    Robot.motorsStop();
    delay(500);
  }

  Robot.stopPlayFile();
  Robot.clearScreen();
}
```

Lista funzioni Arduino utilizzate nella scheda *"Def_funzioni.h"*

- Robot.motorsWrite(*speedLeft, speedRight*);
  La funzione permette di controllare velocità e direzione dei due motori presenti sulla *motor board*. Il primo parametro si riferisce alla ruota sinistra mentre il secondo alla destra. I valori accettati devono essere compresi tra -255 e 255. Il numero 0 permette di fermare i robot; con valori maggiori di zero la ruota si muoverà avanti, con valori minori di 0 la ruota si muoverà con velocità progressiva indietro. Nessun ritorno.

- Robot.motorsStop();
  Permette di arrestare immediatamente entrambi i motori del robot. Nessun parametro e nessun ritorno previsto.

- Robot.clearScreen();
  Riempie tutto lo schermo con il colore di default (bianco). Questo si traduce nel cancellare tutte le scritte o immagini stampate precedentemente. Nessun parametro e nessun ritorno.

- Robot.drawBMP(*filename, x, y*);
  Mostra un immagine sullo schermo. L'immagine deve essere memorizzata nella scheda SD e deve rispettare le seguenti specifiche: non più grande di 128 * 160 px e profondità di colore di 24 bit.
  Se l'immagine supera la dimensione standard, verrà tagliata.
  Con il primo parametro si indica il nome con cui è individuata l'immagine, il secondo e il terzo definisco rispettivamente l'ascissa e l'ordinata di partenza per posizionare all'interno dello schermo l'immagine. Nessun ritorno previsto.

- random(*min, max*);
  La funzione genera numeri pseudo-casuali. Il primo parametro indica il numero minimo da generare mentre il secondo il numero massimo. Il ritorno della funzione è appunto il numero generato di tipo *long* e compreso tra *min* e *max-1*.

- Robot.playFile(*filename*);
  Riproduce un file salvato sulla scheda SD. Attraverso il primo parametro si comunica il nome del file che deve essere rigorosamente in formato *.sqm* e salvato nella SD. Per utilizzare la funzione è necessario inserire nel *void "setup() { …"* le due istruzioni che inizializzino la lettura SD e lo speaker. Nessun ritorno.

- Robot.stopPlayFile();
  L'istruzione serve per arrestare la riproduzione di un file *.sqm*. Nessun parametro e nessun ritorno.