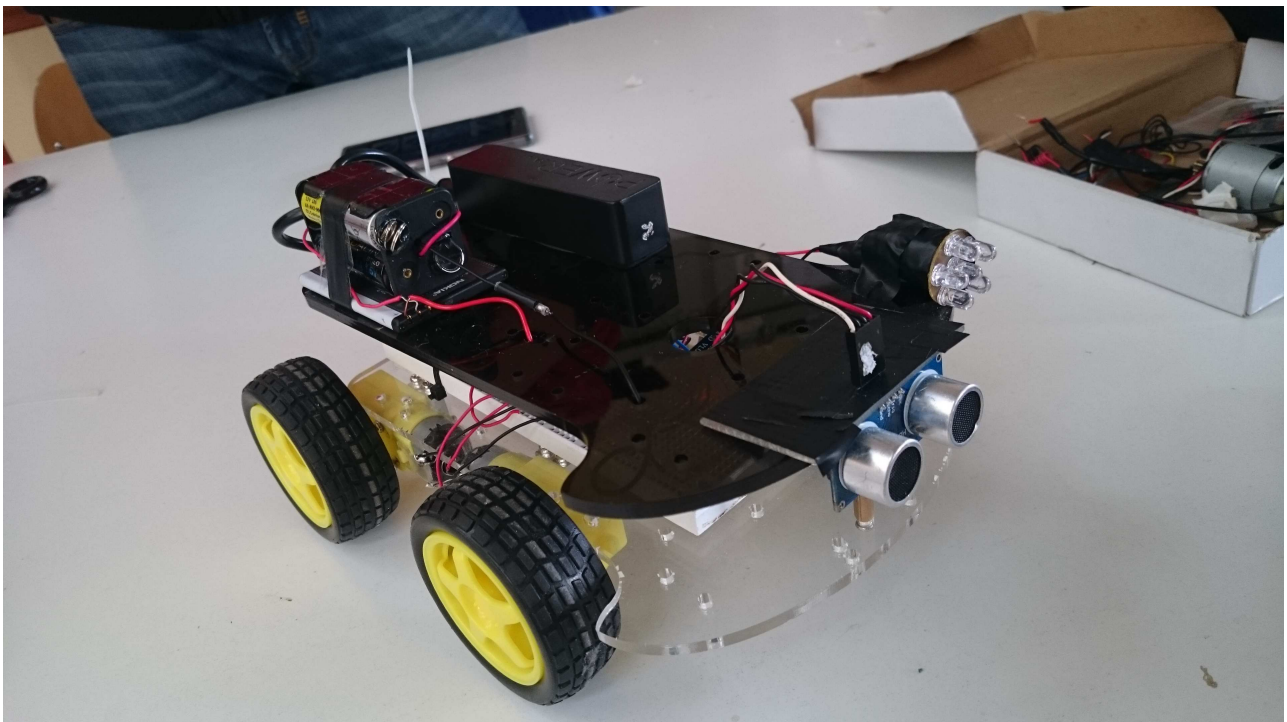




Relazione auto a trazione elettrica comandata con ARDUINO



Powered by:
Ventrella Niccolò
Sciaruto Mattia
Giannarelli Nicola
Valenti Giorgia
Loiacono Lucadomenico

INDICE

- Introduzione.....3
- Materiale utilizzato.....4/5
- Funzionamento.....6
- Realizzazione circuito.....6
- Programmazione Arduino.....7-14
- Possibili miglioramenti.....15
- Collegamenti con le altre materie.....16
- Possibili utilizzi nella vita reale.....16
- Conclusioni e foto del gruppo.....17

INTRODUZIONE

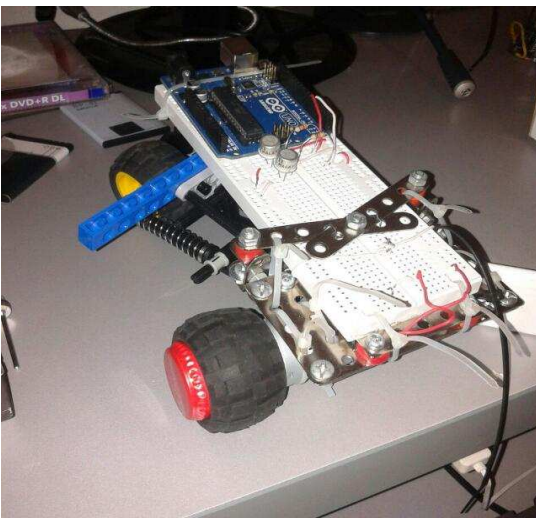
In questa relazione, noi, "Best Electronic Group" della classe 4Q, vi illustreremo come realizzare una macchina guidata da arduino in due differenti modalità:

- Autonomo;
- Teleguidato, mediante uso del bluetooth da applicazioni android.

Il primo prototipo di "Macchina", nasce nel 2013.

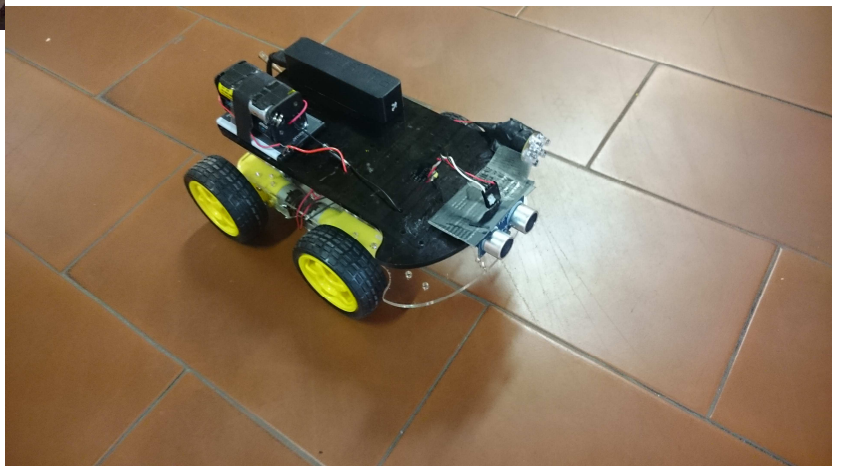
Dopo diversi tentativi di una realizzazione completa, abbiamo deciso di acquistare una struttura già munita di motori e appositi sistemi di bloccaggio, in modo da semplificare il lavoro.

Nel momento in cui ci siamo accorti che la nostra buona scelta, il progetto ha cominciato a prendere forma.



← PRIMA

DOPO ->



MATERIALE UTILIZZATO

Durante il nostro percorso di realizzazione abbiamo utilizzato diversi materiali, alcuni qui sotto elencati:

ARDUINO UNO



Arduino uno è una scheda elettronica di piccole dimensioni, con un microcontrollore e circuiteria di contorno, mediante la quale si possono realizzare molti progetti. L'Arduino necessita la programmazione tramite linguaggio MicroC, un derivato del linguaggio C.

MOTORI DC 6V (x4)



$V_{dc}=6v$

LED



$I=15-17\text{ mA}$

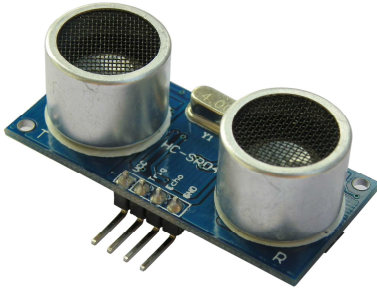
$V=1.2v$

INTEGRATO L293B (PONTE H)

L'integrato L293B TEXAS INSTRUMENTS, comunemente noto come ponte H, è solo uno dei tanti modelli di circuiti di controllo. Il ponte H è un circuito elettronico che può funzionare nei quattro quadranti del piano "corrente-tensione" sul carico. In breve, tramite un'alimentazione esterna (nel nostro caso ricevuta dall' Arduino), il ponte H permette il pieno controllo sulla rotazione dei motori.

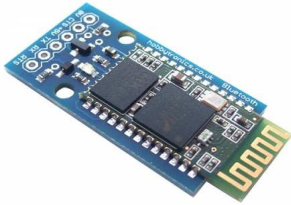


SONAR HC-SR04



HC-SR04 è un sensore a ultrasuoni che misura il tempo impiegato dalle onde sonore emesse da un sorgente, dal momento dell'invio, e, dopo l'incontro di un ostacolo, il loro ritorno. Tramite il tempo calcolato si è in grado di ricavare la distanza.

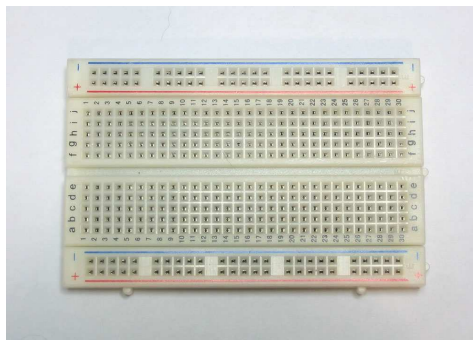
SHIELD BLUETOOTH



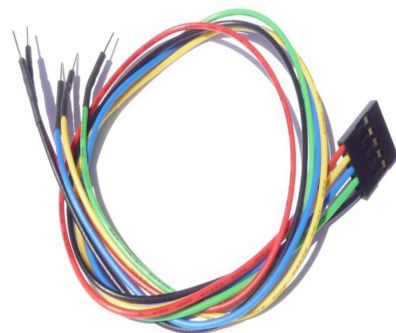
La shield bluetooth è un dispositivo integrativo all'Arduino. Questo piccolo dispositivo permette l'uso, e il controllo dell'arduino da dispositivi bluetooth, precedentemente accoppiati con la shield, chiamata HC-06.

Per utilizzarla è necessario disporre dell'applicazione android "arduino rc controller".

BREADBOARD



CAVI DI RAME RIGIDO (d=0.25mm)



FUNZIONAMENTO

La macchinina presenta diverse funzionalità.

Inizialmente tramite l'utilizzo degli ultrasuoni

“HC-SR04”, il veicolo in movimento riesce ad individuare un ostacolo posto ad una distanza preimpostata, e data una funzione di controllo nel programma lo evita cambiando direzione; la scelta della direzione avviene per mezzo di un controllo, nel quale la macchina compie una rotazione di 90° a destra e 90° a sinistra dell'ostacolo, e si dirige sulla via dove rileva la profondità maggiore.

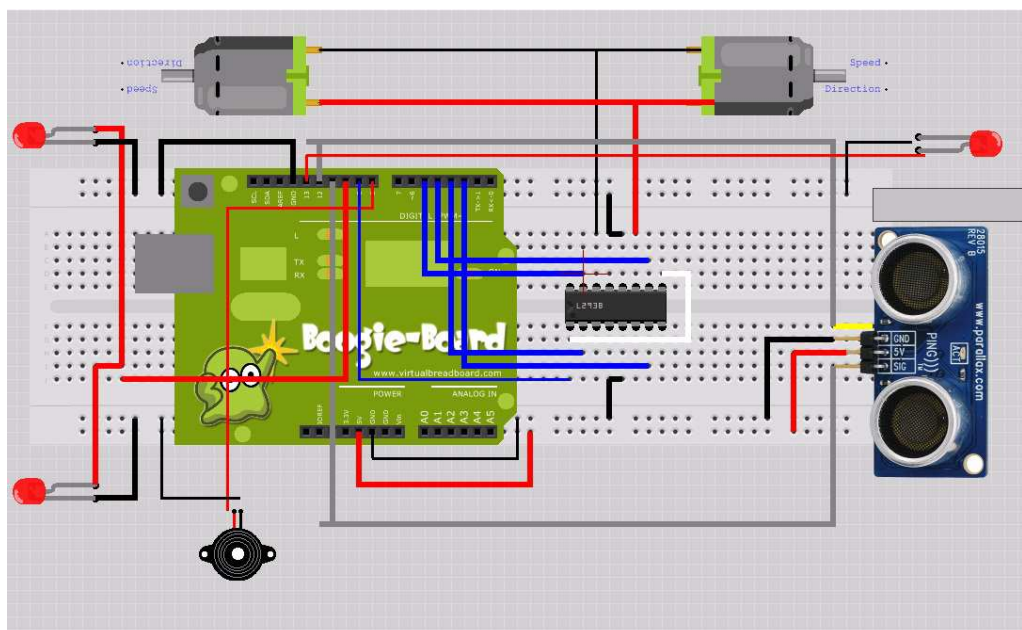
Secondariamente la macchinina può essere guidata mediante l'uso di un applicazione scaricabile da play store per soli smartphone android, che utilizza la connessione bluetooth, connettendosi alla shield bluetooth (HC-06) integrata sul veicolo. L'utilizzo si estende anche alla possibilità di utilizzare un impianto sonoro tramite uno speaker e luminoso tramite vari led piazzati sulla struttura.

PROCEDIMENTO OPERATIVO

La realizzazione del progetto è avvenuta in 2 diverse fasi:

- Realizzazione del circuito;
- Programmazione dell'arduino.

REALIZZAZIONE DEL CIRCUITO



PROGRAMMAZIONE DELL'ARDUINO

```
#include <SoftwareSerial.h>
#define C 2100
#define D 1870
#define E 1670
#define f 1580
#define G 1400
#define R 0

int bluetoothTx = 0; //Trasmittitore bluetooth
int bluetoothRx = 1; //Ricevitore bluetooth
const int motorPin1 = 2; //Motore destro massa
const int motorPin2 = 3; //Motore destro positivo
const int motorPin3 = 4; //Motore sinistro positivo
const int motorPin4 = 5; //Motore sinistro massa
int speakerOut = 8; //Speaker
const int enablePin = 9; //Pin di accensione ponte H
const int led1 = 10; //Luci posteriori
int echoPin = 11; //Ricevitore ad infrarossi
int trigPin = 12; //Trasmittitore ad infrarossi
const int led = 13; //Luci anteriori

char lettera; //Assegnamento da bluetooth
int cont = 0; //Contatore per automatizzare la macchina
int contA = 0; //Contatore note
int contB = 0;

SoftwareSerial bluetooth(blueToothTx, blueToothRx);

void setup()
{
  //Dichiaro i componenti di OUTPUT e INPUT
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  pinMode(motorPin1, OUTPUT);
  pinMode(motorPin2, OUTPUT);
  pinMode(motorPin3, OUTPUT);
  pinMode(motorPin4, OUTPUT);
  pinMode(enablePin, OUTPUT);
  pinMode(led, OUTPUT);
  pinMode(led1, OUTPUT);
```

```
pinMode(speakerOut, OUTPUT);

Serial.begin(9600);      //Avvio monitor seriale (se connesso a PC)
bluetooth.begin(115200); //Avvio la comunicazione tra bluetooth e PC/dispositivo
bluetooth.print("$$$"); //Scrivo su monitor seriale "$$$"
delay(100);             //Pausa di 100mS
bluetooth.println("U,9600,N"); //Scrivo su monitor seriale del dispositivo bluetooth (dove possibile)
bluetooth.begin(9600);  //Avvio la comunicazione tra bluetooth e PC/dispositivo

digitalWrite(enablePin, HIGH); //Accendo il ponte H

}

//LOOP BLUETOOTH
void loop()
{

  if(contA==58)
  {
    contA=0;    //Azzera il contatore note
  }

  if(bluetooth.available()) //Se un dispositivo bluetooth e'connesso
  {
    do{
      lettera = (char)bluetooth.read(); //lettera' assume il valore dato da bluetooth
      Serial.print(lettera);           //Scrive su monitor seriale 'lettera'

      if(lettera=='W')
      {
        ledON();           //Accende le luci anteriori
      }

      if(lettera=='w')
      {
        ledOFF();         //Spegne le luci anteriori
      }

      if(lettera=='U')
      {
        led1ON();         //Accende le luci posteriori
      }
    }
  }
}
```

```
    if(lettera=='u')
    {
        led1OFF();          //Spegne le luci posteriori
    }

    if(lettera=='X')
    {
        cont=1;            //Incremento 'cont'
    }

    if(lettera=='x')
    {
        cont=0;            //Decremento 'cont'
    }

    if(cont==1)
    {
        automatica();      //Funzione per automatizzare la macchina
    }
}while(cont==1);          //Ripete il ciclo affinche 'cont' ha valore 1

if(lettera=='F')
{
    avanti();    //Funzione avanti
}

if(lettera=='B')
{
    indietro(); //Funzione indietro
}

if(lettera=='S')
{
    ferma();    //Funzione ferma
}

if(lettera=='L')
{
    sinistra(); //Funzione sterzo a sinistra
}

if(lettera=='R')
{
    destra();   //Funzione sterzo a destra
```

```

    }
}

if(lettera=='V')
{
    contB=1;
}

if(lettera=='v')
{
    contB=0;
}

if(contB==1) // LET THE WILD RUMPUS BEGIN =====
{
int melody[] = {E, E, E,R,
E, E, E,R,
E, G, C, D, E, R,
f, f, f,f, f, E, E,E, E, D ,D,E, D, R, G ,R,
E, E, E,R,
E, E, E,R,
E, G, C, D, E, R,
f, f, f,f, f, E, E, E, G,G, f, D, C,R };
int MAX_COUNT = sizeof(melody) / 2; // Melody length, for looping.
// Set overall tempo
long tempo = 5000;
// Set length of pause between notes
int pause = 500;
// Loop variable to increase Rest length
int rest_count = 100; //<-BLETCHEROUS HACK; See NOTES
// Initialize core variables
int tone_ = 0;
int beat = 0;
long duratin = 0;

tone_ = melody[contA];
beat = 50;

duratin = beat * tempo; // Set up timing

playTune(tone_,rest_count,duratin);
// A pause between notes...
delayMicroseconds(pause);

```

```

contA++;
}
}

```

```

void automatica()
{
  int d,s;          //Variabili per controllo distanza laterale
  long distance;    //Variabile per controllo distanza anteriore
  distance=distanza(); //distance' viene calcolata e restituita nella funzione 'distanza()'
  monitorSeriale(distance); //Scrive su monitor seriale 'distance'
  if(distance>25)
  {
    avanti();

  }
  else
  {
    ferma();
    delay(500);
    controllo(d,s); //Funzione per controllo distanza laterale, al quale passiamo le apposite variabili 'd', ed 's'
  }
}

```

```
//CONTROLLO AUTOMATICA
```

```

long distanza()
{
  long duration, distance;
  digitalWrite(trigPin, LOW); //Tramite infrarossi la macchina e' in
  delayMicroseconds(2); //grado di calcolare il tempo da quando
  digitalWrite(trigPin, HIGH); //invia e riceve il segnale ad IR, e
  delayMicroseconds(10); //calcolare la distanza da ciò che ha davanti
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = (duration/2) / 29.1;
  return distance; //Restituisce al programma principale 'distance'
}

```

```

void controllo(int d, int s) //Controllo prossimità laterale
{
  destra(); //Gira a destra per il tempo stabilito
}

```

```
delay(1250);
ferma();          //Si ferma per il tempo stabilito
delay(30);
d=distanza();    //'d' viene calcolata e restituita nella funzione 'distanza()'
delay(30);
sinistra();      //Gira a sinistra per il tempo stabilito
delay(2500);
ferma();          //Si ferma per il tempo stabilito
delay(30);
s=distanza();    //'s' viene calcolata e restituita nella funzione 'distanza()'
delay(30);

if(d>s)          //Se 'd' è maggiore di 's', e quindi la distanza a destra della macchina è più conveniente
{
  destra();      //La macchina gira a destra per il tempo stabilito e sceglie quella direzione
  delay(2500);
}
}

//FUNZIONI MOTORI
void avanti()
{
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, HIGH);
  digitalWrite(motorPin3, LOW);
  digitalWrite(motorPin4, HIGH);
}

void indietro()
{
  digitalWrite(motorPin1, HIGH);
  digitalWrite(motorPin2, LOW);
  digitalWrite(motorPin3, HIGH);
  digitalWrite(motorPin4, LOW);
}

void ferma()
{
  digitalWrite(motorPin1, LOW);
  digitalWrite(motorPin2, LOW);
  digitalWrite(motorPin3, LOW);
  digitalWrite(motorPin4, LOW);
}
```

```
void sinistra()
{
    digitalWrite(motorPin1, LOW);
    digitalWrite(motorPin2, HIGH);
    digitalWrite(motorPin3, HIGH);
    digitalWrite(motorPin4, LOW);
}

void destra()
{
    digitalWrite(motorPin1, HIGH);
    digitalWrite(motorPin2, LOW);
    digitalWrite(motorPin3, LOW);
    digitalWrite(motorPin4, HIGH);
}

//ALTRE FUNZIONI
void monitorSeriale(long distance)
{
    Serial.print("Distanza = ");
    Serial.print(distance);
    Serial.print("\n");
}

void ledON()
{
    digitalWrite(led, HIGH);
}

void led1ON()
{
    digitalWrite(led1, HIGH);
}

void ledOFF()
{
    digitalWrite(led, LOW);
}

void led1OFF()
{
    digitalWrite(led1, LOW);
}
```

```
}

void playTune(int tone_, int rest_count, long duratin)
{

    // PLAY TONE =====
    // Pulse the speaker to play a tone for a particular duration

    long elapsed_time = 0;
    if (tone_ > 0)
    { // if this isn't a Rest beat, while the tone has
        // played less long than 'duration', pulse speaker HIGH and LOW
        while (elapsed_time < duratin)
        {
            digitalWrite(speakerOut,HIGH);
            delayMicroseconds(tone_ / 2);
            // DOWN
            digitalWrite(speakerOut, LOW);
            delayMicroseconds(tone_ / 2);
            // Keep track of how long we pulsed
            elapsed_time += (tone_);
        }
    }
    else
    { // Rest beat; loop times delay

        for (int j=0; j < rest_count; j++)
        { // See NOTE on rest_count
            delayMicroseconds(duratin);
        }
    }
}
```

POSSIBILI MIGLIORAMENTI



Uno dei possibili miglioramenti è dato dall'aggiunta di un particolare integrato comunemente chiamato "MAG". Grazie a questo integrato, la macchinina sarebbe in grado, tramite il rilevamento di campi magnetici, di costruire una mappa di un certo edificio, munito di ferrite (dischi magnetici), stabilendo con ciò i percorsi più brevi per raggiungere un punto B partendo da un punto A.

Un altro possibile miglioramento sarebbe portato dall'aggiunta di una "shield WI-FI", con la quale la macchinina sarebbe in grado di essere controllata da tutti i dispositivi connessi alla sua stessa rete, o da un apposito sito.



Quest'ultima modifica avrebbe il compito di filmare l'andamento della macchina, tramite una telecamera montata sopra al mezzo stesso, in modo da ricreare la grafica di un gioco, dunque la visione in prima persona del viaggio. Tutto questo tramite un monitor al quale sarà collegata la webcam.

COLLEGAMENTI CON ALTRE MATERIE

INFORMATICA: La programmazione dell'arduino è stata fatta in linguaggio "Micro-C", che come il "C++" utilizzato a informatica è un derivato del linguaggio "C".

Precedentemente abbiamo inserito il nostro "sketch", nel quale si può notare questa somiglianza.

MATEMATICA: Per quanto riguarda questa materia, è stato necessario Effettuare dei calcoli per la rotazione della macchinia quando è utilizzata in modalità automatica, in modo tale da regolare gli angoli di rotazione.

INGLESE: La materia inglese è stata coinvolta nella programmazione, in quanto la maggior parte dei comandi sono in lingua inglese.

POSSIBILI UTILIZZI NELLA VITA REALE

Questo prototipo ha diverse funzionalità che potrebbero essere utilizzate nella vita quotidiana.

- ***Robot di salvataggio***: in questa modalità con l'integrazione di una videocamera il prototipo potrebbe esplorare nei casi di calamità naturali, i luoghi dove potrebbe servire sia un aiuto, sia una verifica della sicurezza dell'area.
- ***Guida Turistica***: con l'ausilio dello speaker, con all'interno una voce pre-registrata, il prototipo, tramite controllo da smartphone, risulterebbe utile a guidare i turisti nelle tappe più importanti delle diverse città.
- ***Guida per i cechi***: con l'aggiunta del MAG e relative mappatura di un edificio, la macchinina potrebbe guidare un cieco nella sua vita quotidiana, e con l'aggiunta di messaggi sonori che grazie allo speaker, andrebbero ad indentificare l'area in cui l'umano si trova.
- ***Robot Casalingo***: con l'applicazione di 2 semplici spazzole nella parte anteriore del mezzo, il prototipo potrebbe effettuare la funzione di una scopa per le pulizie domestiche.

CONCLUSIONI

La realizzazione del progetto è stata lunga e piena di imprevisti, tuttavia il risultato è estremamente soddisfacente.

Allegati alla relazione sono stati realizzati 2 video:

- Video del progresso temporale del prototipo;
- Video con rappresentazione del comando del prototipo da smartphone.



Eccoci, i componenti di “BEST ELECTRONIC GROUP”.

