

Automobile Bluetooth

Ivancich Stefano

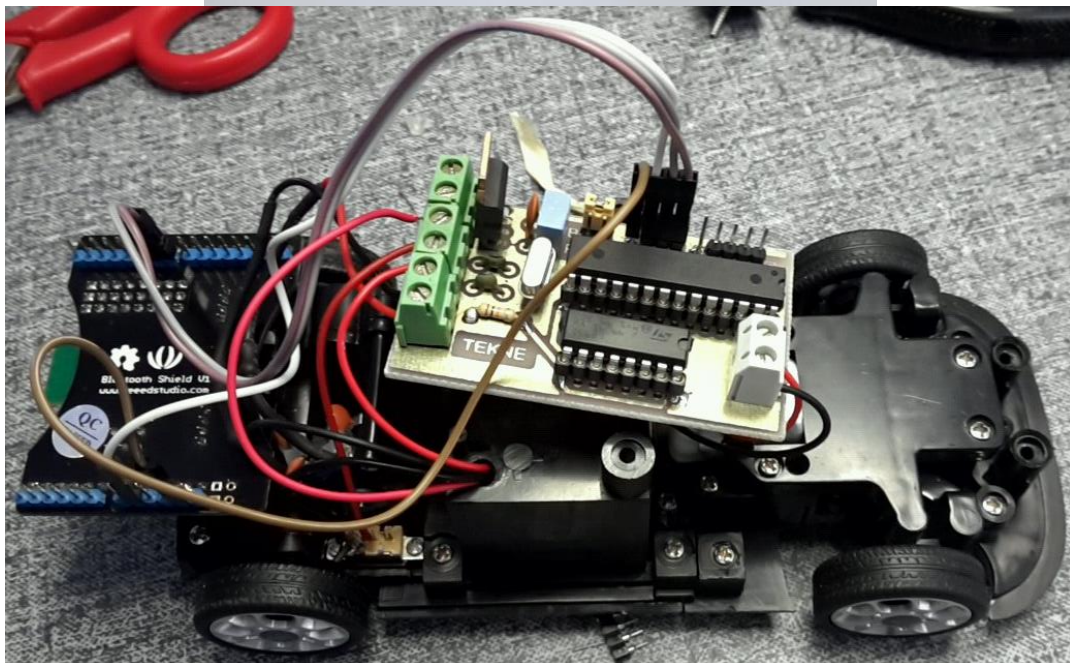
17 Maggio 2015

SOMMARIO

Di seguito s'illustra un progetto di un'automobile comandata da uno smartphone tramite la connessione Bluetooth.

E' stata presa una automobilina RC Fiat 500 (7cm di larghezza x 20 cm di lunghezza x 6 cm di altezza) da cui li è stata tolta completamente l'elettronica interna ed è stata sostituita da quella da me progettata.

Questo progetto integra al suo interno il microcontrollore ATmega328p.



INDICE

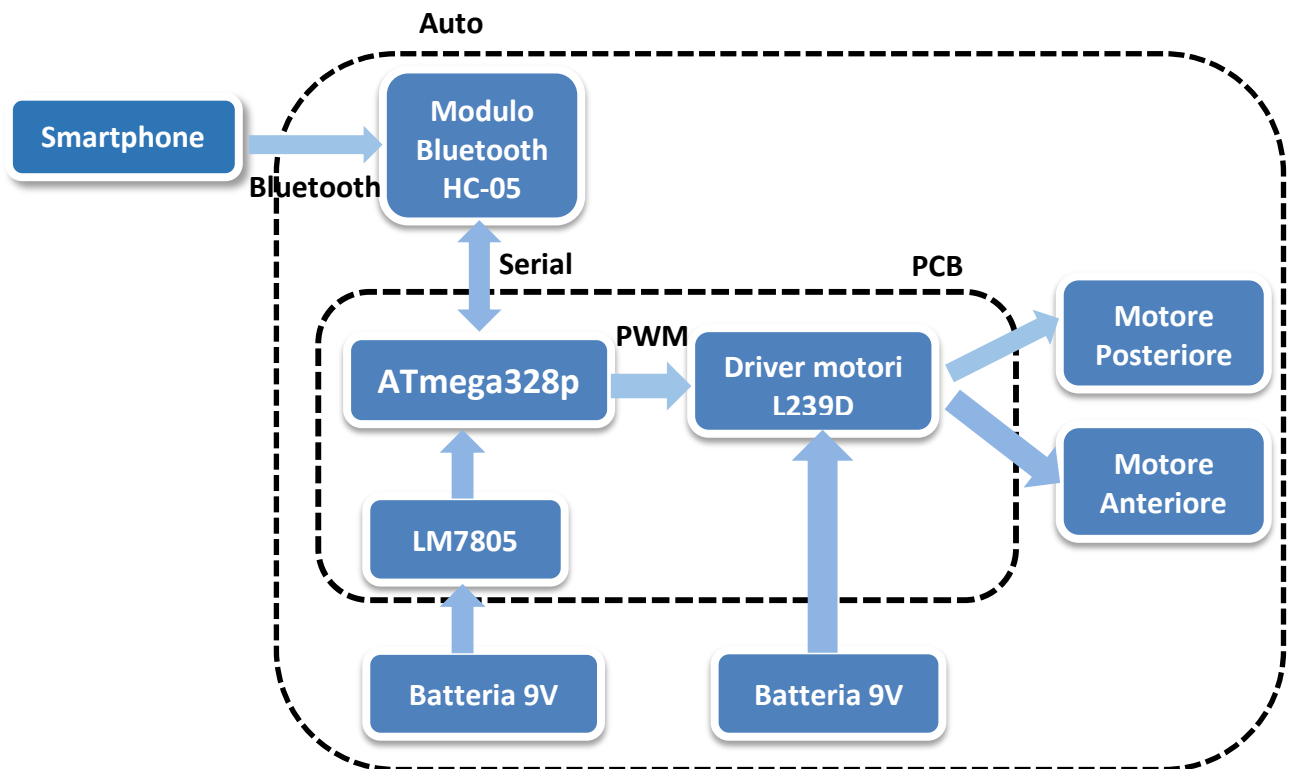
SOMMARIO.....	
INTRODUZIONE.....	1
SCHEMA A BLOCCHI AUTOMOBILE	1
SCHEMA ELETTRICO	1
PARTI DEL PROGETTO	3
Alimentazione.....	3
LM7805	3
Modulo Bluetooth HC-05.....	3
Microcontrollore AtMega328p.....	4
Driver motori L293D	4
Motori	5
PCB.....	6
SOFTWARE	7
LISTA COMPONENTI.....	10
CONCLUSIONI	10

INTRODUZIONE

L'obiettivo di questo progetto è realizzare un'automobile di ridotte dimensioni comandata tramite smartphone.

L'automobile è costituita da: 2 batterie a 9V, 2 motori in continua, un modulo Bluetooth HC-05, un AtMega328p, un regolatore di tensione LM7805 e un driver motori L293D.

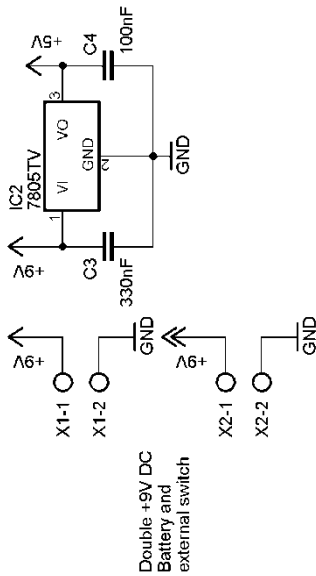
SCHEMA A BLOCCHI AUTOMOBILE



SCHEMA ELETTRICO

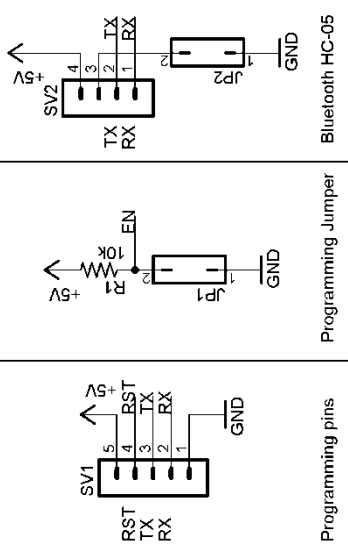
SUPPLY

7805 Voltage regulator



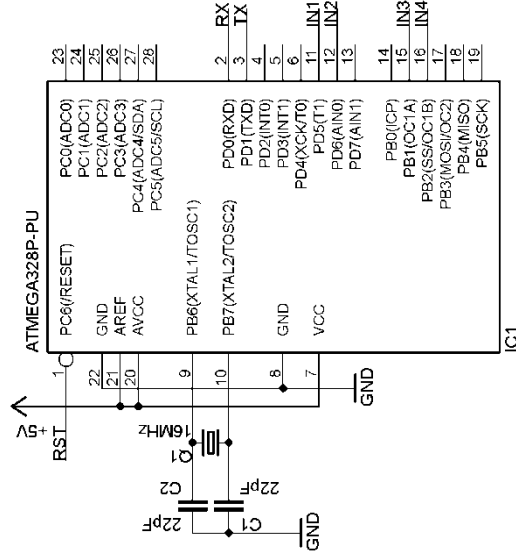
INTERFACE

Serial communication for programming and bluetooth



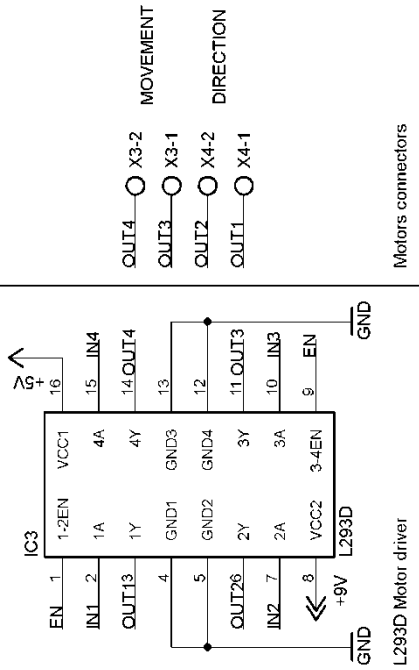
CORE

ATmega328P base configuration



MOTORS

Motor driver and connectors



Ivancich Stefano
ITIS C. Zuccante
Progetto TEKNE

TekneCar
06/05/2015 17:43:40
Sheet: 1/1

PARTI DEL PROGETTO

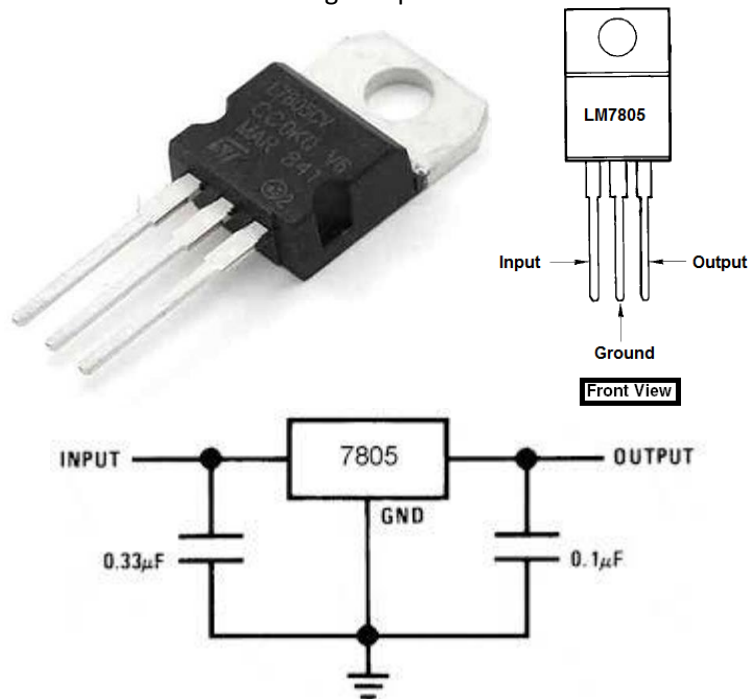
Alimentazione

L'alimentazione è separata per la logica e per i motori, ed è fornita da due batterie da 9V. Questo perché i motori in fase di avvio o di stallo richiedono talmente tanta corrente da doverla sottrarre all'ATmega328p, quindi spegnendolo e di conseguenza fermando l'automobile.



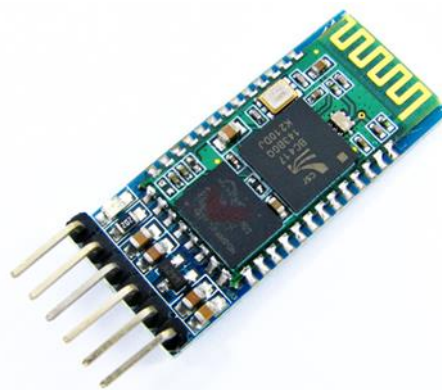
LM7805

Il regolatore di tensione LM7805 prende in ingresso la tensione di 9V della batteria e in uscita fornisce 5V necessari all'alimentazione dell'ATMega328p e al modulo Bluetooth.



Modulo Bluetooth HC-05

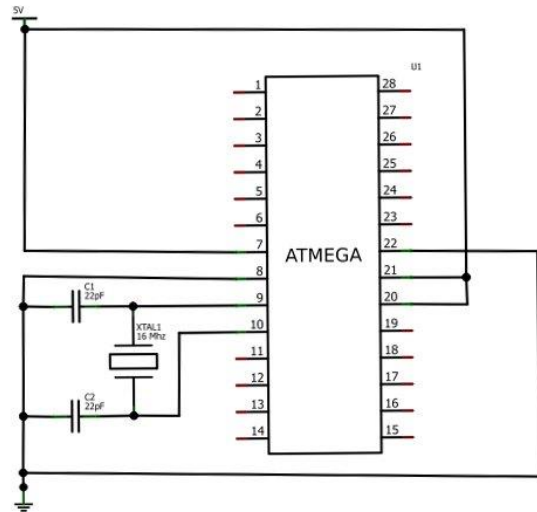
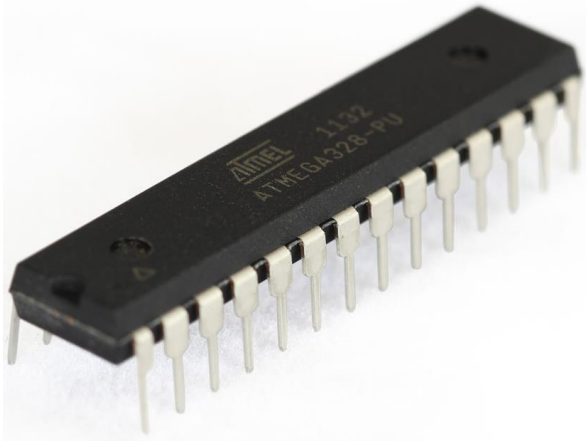
Questa scheda permette un dialogo con dispositivi Bluetooth tramite comunicazione seriale. Ha una portata di 10mt, si imposta facilmente tramite comandi AT ed è programmabile sia come master che come slave. Accetta alimentazione e comandi sia a 5Vdc che a 3,3Vdc.



Microcontrollore AtMega328p

L'ATMega328p è il microcontrollore utilizzato dall'Arduino UNO, questo permette di usare codice Arduino in un progetto senza avere una intera e ingombrante scheda.

Per utilizzare l'ATMega328p sono necessari un oscillatore al quarzo da 16Mhz, una tensione di alimentazione di 5V e una connessione seriale.



ATmega328 Pin Mapping

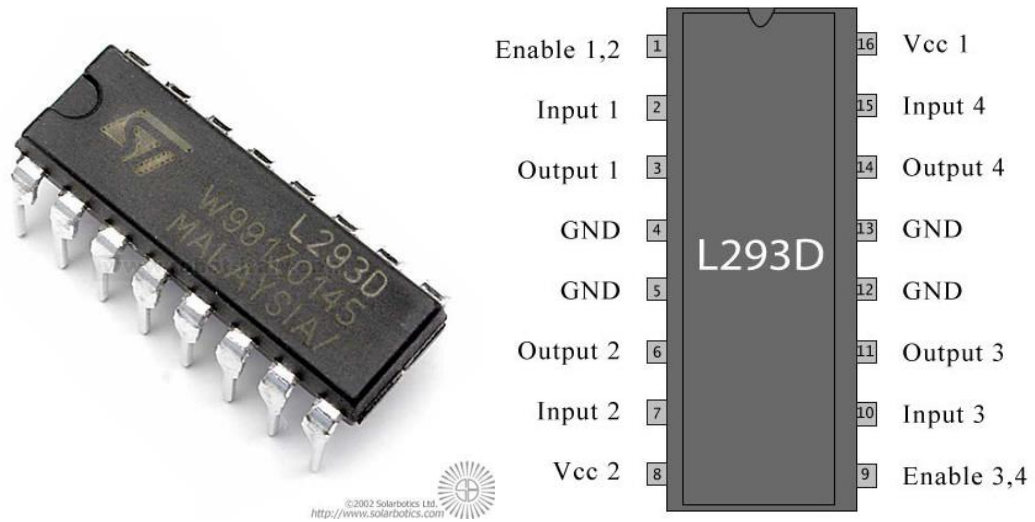
Arduino function	ATmega328P Pin	ATmega328P Pin	ATmega328P Pin	Arduino function
reset	(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13) analog input 5
digital pin 0 (RX)	(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12) analog input 4
digital pin 1 (TX)	(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11) analog input 3
digital pin 2	(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10) analog input 2
digital pin 3 (PWM)	(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9) analog input 1
digital pin 4	(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8) analog input 0
VCC	VCC	7	22	GND GND
GND	GND	8	21	AREF analog reference
crystal	(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC VCC
crystal	(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5) digital pin 13
digital pin 5 (PWM)	(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4) digital pin 12
digital pin 6 (PWM)	(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3) digital pin 11 (PWM)
digital pin 7	(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2) digital pin 10 (PWM)
digital pin 8	(PCINT0/CLKO/CP1) PB0	14	15	PB1 (OC1A/PCINT1) digital pin 9 (PWM)

Digital Pins 11, 12 & 13 are used by the ICSP header for MISO, MOSI, SCK connections (Atmega 168 pins 17, 18 & 19). Avoid low-impedance loads on these pins when using the ICSP header.

Driver motori L293D

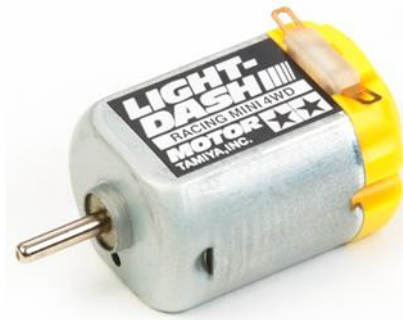
Può pilotare indipendentemente una coppia di motori CC da 0.6 A (max) con voltaggio da 4.5 a 36 volt.

E' dotato di una protezione interna che, in caso di surriscaldamento, limita la corrente in uscita fino al ripristino della temperatura ottimale.



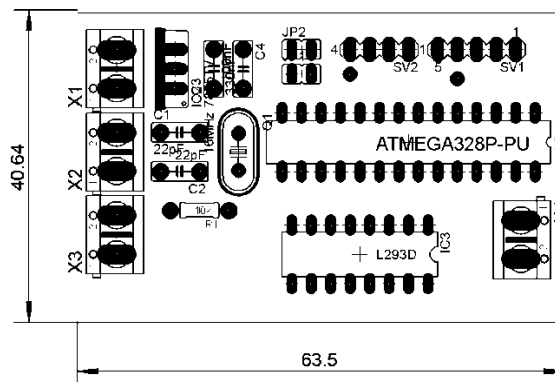
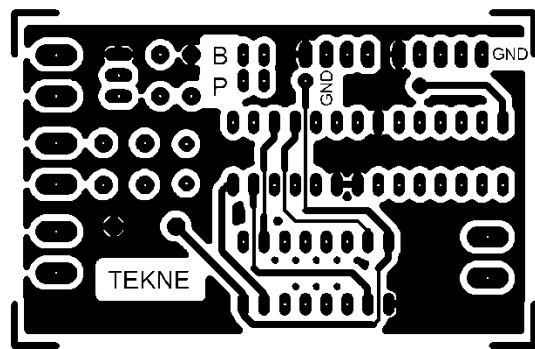
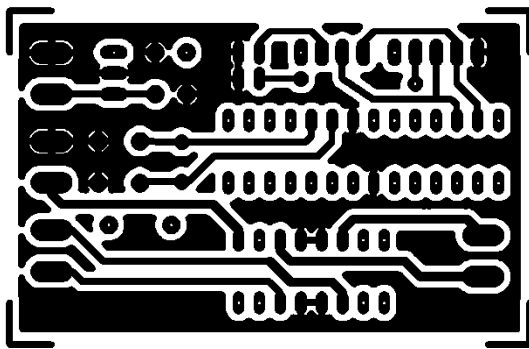
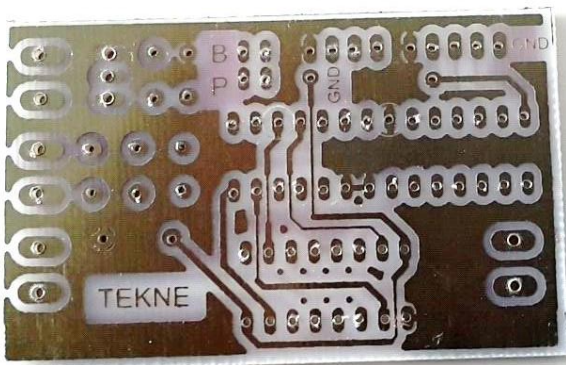
Motori

Sia il motore posteriore che quello anteriore sono i motorini dell'automobile originale, che sono dei semplici motorini in continua delle mini 4WD. Vengono alimentati dalla batteria a 9V a loro dedicata.



PCB

Il pcb e lo schema elettrico sono stati realizzati mediante l'utilizzo del software EAGLE CAD 7.1



SOFTWARE

La macchinina deve quindi ricevere via Bluetooth dal telefono i comandi necessari per muoversi. Ovvero avanti, indietro, destra, sinistra, fermo, sono stati aggiunti i comandi open Bluetooth connection, close Bluetooth connection, e aumenta velocità, perché scaricandosi la batteria i motori girano più lentamente quindi per ottenere la velocità originale è necessario aumentare il PWM di ciascuna direzione.

Sono stati previsti i 5 pin di programmazione (VCC, GND Reset, RX, TX) in modo tale da non staccare l'ATmega328p dal PCB.

```
////////////////////// DICHIARAZIONE PIN ////////////////////////////////////////
int MotFrontA=5; // Motore Davanti A
int MotFrontB=6; // Motore Davanti B
int MotBackA=9; // Motore Dietro A
int MotBackB=10; // Motore Dietro B

////////////////////// VARIABILI ////////////////////////////////////////
int velAvanti=70;
int velIndietro=70;
int velDestra=80;
int velSinistra=80;
int velSpinta=130;

boolean enableCommunication=false; // Flag che serve ad abilitare e disabilitare
lo scambio di dati
String stringa=""; // Stringa ricevuta via bluetooth

////////////////////// IMPOSTAZIONI ////////////////////////////////////////
void setup()
{
// PIN MOTORI:
pinMode(MotFrontA, OUTPUT); // Motore Davanti A
pinMode(MotFrontB, OUTPUT); // Motore Davanti B
pinMode(MotBackA, OUTPUT); // Motore Dietro A
pinMode(MotBackB, OUTPUT); // Motore Dietro B

setupBluetoothConnection();
}

void loop()
{
char recvChar; // Variabile che contiene il byte ricevuto via bluetooth
if(Serial.available()){ //check if there's any data sent from the remote
bluetooth shield
recvChar = Serial.read(); // Leggo 1 byte dallo stack di dati ricevuti
if(recvChar=='c'){
enableCommunication=false; // Per disabilitare lo scambio di dati
setupBluetoothConnection();
}
if(enableCommunication){ // Se lo scambio di dati è abilitato
stringa="";
stringa.concat(recvChar);
while(Serial.available()){
recvChar = Serial.read();
stringa.concat(recvChar);
}
if(stringa=="a") AVANTI();
if(stringa=="d") DESTRA();
if(stringa=="s") SINISTRA();
if(stringa=="i") INDIETRO();
if(stringa=="f") FERMO();
}
```

```

        if(stringa=="u") AUMENTA();
    }
    if(recvChar=='o') enableCommunication=true; // Per abilitare lo scambio di
dati
    }
}

void AUMENTA ()
{
    velAvanti=velAvanti+10;
    velIndietro=velIndietro+10;
    velDestra=velDestra+10;
    velSinistra=velSinistra+10;
    velSpinta=velSpinta+10;
}

void AVANTI ()
{
    digitalWrite(MotFrontA, LOW);
    digitalWrite(MotFrontB, LOW);
    digitalWrite(MotBackA, LOW);
    analogWrite(MotBackB, velSpinta);
    delay(300);
    analogWrite(MotBackB, velAvanti);
}

void DESTRA ()
{
    digitalWrite(MotFrontA, HIGH);
    digitalWrite(MotFrontB, LOW);
    digitalWrite(MotBackA, LOW);
    analogWrite(MotBackB, velSpinta);
    delay(300);
    analogWrite(MotBackB, velDestra);
}

void SINISTRA ()
{
    digitalWrite(MotFrontA, LOW);
    digitalWrite(MotFrontB, HIGH);
    digitalWrite(MotBackA, LOW);
    analogWrite(MotBackB, velSpinta);
    delay(300);
    analogWrite(MotBackB, velSinistra);
}

void INDIETRO ()
{
    digitalWrite(MotFrontA, LOW);
    digitalWrite(MotFrontB, LOW);
    analogWrite(MotBackA, velSpinta);
    delay(300);
    analogWrite(MotBackA, velIndietro);
    digitalWrite(MotBackB, LOW);
}

void FERMO ()
{
    digitalWrite(MotFrontA, LOW);
    digitalWrite(MotFrontB, LOW);
    digitalWrite(MotBackA, LOW);
    digitalWrite(MotBackB, LOW);
}

```

```

}

void setupBlueToothConnection()
{
  Serial.begin(38400); // Set BluetoothBee BaudRate to
  default baud rate 38400
  Serial.print("\r\n+STWMOD=0\r\n"); // set the bluetooth work in
  slave mode
  Serial.print("\r\n+STNA=ArduinoBluetooth\r\n"); // set the bluetooth name
  Serial.print("\r\n+STOAUT=1\r\n"); // Permit Paired device to
  connect me
  Serial.print("\r\n+STAUTO=0\r\n"); // Auto-connection should be
  forbidden here
  delay(2000); // This delay is
  required.
  Serial.print("\r\n+INQ=1\r\n"); // make the slave bluetooth
  inquirable
  Serial.println("The slave bluetooth is inquirable!");
  delay(2000); // This delay is
  required.
  Serial.flush();
}

```

LISTA COMPONENTI

Componente	Q/TA
Macchinina Fiat 500 1:20 (19cmx7cmx8cm)	1
Modulo Bluetooth HC-05	1
Regolatore di tensione LM7805	1
Batteria 9V	2
ATmega328p	1
Strip tulipano	44
Resistenza 10k Ω	1
Strip maschio	13
Condensatore ceramico 330Nf	1
Condensatore ceramico 22pF	2
Oscillatore al quarzo 16MHz	1
Jumper	1
Driver motori L293D	14
Connettore 2 poli	4
Condensatore plastico 100nF	1

CONCLUSIONI

Inizialmente è stata progettata un'altra basetta elettronica con l'alimentazione per l'ATmega328p e per i motori comune. Ma i motori quando andavano in stallo richiedevano talmente tanta corrente da doverla sottrarre all'ATmega328p, quindi spegnendolo e di conseguenza fermando l'automobile.

Sfortunatamente non è stato possibile reperire un modulo Bluetooth hc-05 per l'automobile ma è stata usata una shield Bluetooth compatibile.