

A.S.L. IN COLLABORAZIONE CON F.C.A  
A.S. 2018/19 CLASSE 5C

# **IMPIANTO SEMAFORICO AGV**

# IL GRUPPO DI LAVORO



# INTRODUZIONE

- ✘ La presente analisi tecnica fa riferimento al relativo documento di analisi funzionale:
  - + ASL5C.PDF



The image shows the cover of a document titled "Analisi funzionale" (Functional Analysis). The cover features the logos of "IIS PRIMO LEVI" and "FCA FIAT CHRYSLER AUTOMOBILES". The main title is "ALTERNANZA SCUOLA LAVORO Sistema semaforico per traffico AGV". Below the title, it says "Analisi funzionale". At the bottom, it indicates "Classe 5°C I.I.S. Primo Levi A.S. 2018/19". There are several orange circles of varying sizes on the left side of the cover, with the number "1" inside one of them.

 **IIS PRIMO LEVI**

  
FIAT CHRYSLER AUTOMOBILES

**ALTERNANZA SCUOLA LAVORO**  
**Sistema semaforico per**  
**traffico AGV**

**Analisi funzionale**

Classe 5°C I.I.S. Primo Levi A.S. 2018/19

# COMUNICAZIONE BLUETOOTH- IL CIRCUITO

---

# COMUNICAZIONE BLUETOOTH-MATERIALE

- ✘ SCHEDA BLUETOOTH MASTER HC-05
  - + COLLEGAMENTO CON SCHEDA ARDUINO MEGA
- ✘ SCHEDA BLUETOOTH SLAVE HC-06
  - + COLLEGAMENTO CON SCHEDA ARDUINO GENUINO

# COMUNICAZIONE BLUETOOTH- INIZIALIZZAZIONE SCHEDA MASTER

---

- ✘ Occorre inizializzare la scheda master in modo che si predisponga ad inviare messaggi
- ✘ Premere alcune volte il pulsante sulla scheda bluetooth quando non è collagata
- ✘ Tenere premuto il pulsante prima di dare corrente alla scheda tramite VCC
- ✘ Quando il led sulla scheda lampeggia, posso caricare il codice arduino ed inviare i comandi dal monitor seriale

# COMUNICAZIONE BLUETOOTH- INIZIALIZZAZIONE SCHEDA MASTER – CODICE ARDUINO

```
|
//libreria per la comunicazione fra le due schede bluetooth
#include <SoftwareSerial.h>
//oggetto utilizzato per la comunicazione fra le due schede
SoftwareSerial BTserial(10, 11 ); // RX | TX
// Connettere HC-05 TX al pin Arduino 10 (RX).
// Connettere HC-05 RX al pin Arduino 11 TX con un circuito apposito che
// abbassi la corrente dai 5V di arduino ai 3.3V della scheda

char c = ' ';

void setup()
{
  Serial.begin(9600);
  Serial.println("Arduino is ready");
  Serial.println("Remember to select Both NL & CR in the serial monitor");

  // HC-05 default serial speed for AT mode is 38400
  //SOLO IN FASE DI SET-UP IMPOSTARE 38400 BAUD
  //PER L'AT MODE DI SETUP DEI COMANDI
  BTserial.begin(38400);
}
```

# COMUNICAZIONE BLUETOOTH- INIZIALIZZAZIONE SCHEDA MASTER – CODICE ARDUINO

```
void loop()
{

    // leggo dal Serial Monitor su PC collegato ad arduino
    //e invio alla scheda bluetooth HC-05
    if (Serial.available())
    {
        c = Serial.read();
        BTserial.write(c);
    }

    // Leggo dalla scheda bluetooth HC-05 invio al Serial Monitor sul pc collegato ad arduino
    if (BTserial.available())
    {
        c = BTserial.read();
        Serial.write(c);
    }

}
```

# COMUNICAZIONE BLUETOOTH- INIZIALIZZAZIONE SCHEDA MASTER - COMANDI DA MONITOR SERIALE SU PC A SCHEDA BLUETOOTH

```
//ELENCO DEI COMANDI PER IL SET-UP DA LANCIARE DAL PC  
//AT per controllare di essere in AT-MODE- restituisce ok  
//AT+ROLE=1 per impostare la modalità master- restituisce ok  
//AT+CMODE=1 per fare in modo che si colleghi a qualsiasi scheda bluetooth  
//- nel nostro caso alla scheda slave HC-06
```

# CODICE SCHEDA PIR/MASTER – CODICE COMPLETO

```
//CODICE SCHEDA MASTER|
//pin del sensore PIR che rileva il movimento del passaggio del carrello AGV
#define pir 4
//libreria per la connessione via bluetooth tra due schede bluetooth
#include <SoftwareSerial.h>

//HC-05 è la scheda master qui gestita per inviare un messaggio
SoftwareSerial BTserial(2, 3); // RX | TX

//il messaggio inviato è '1' per dire all'altra scheda di accendere il rosso
char isRed = '0';

//variabile per testare se sta passando il carrello AGV
//LOW, non sta passando; HIGH, sta passando il carrello
int isPassing = LOW;
```

# CODICE SCHEDA PIR/MASTER – CODICE COMPLETO

```
/**inizializzazione  
void setup() {  
    //inizializzo la connessione tra le schede bluetooth in modalità seriale  
    //usare lo stesso baud rate su master e slave  
    BTserial.begin(9600);  
  
}
```

# CODICE SCHEDA PIR/MASTER – CODICE COMPLETO

```
/***/programma di gestione SENSORE PIR e scheda BLUETOOTH MASTER
void loop() {
  //leggo se sta passando il carrello
  isPassing = digitalRead(pir);

  //se il sensore PIR rileva un carrello
  if (isPassing == HIGH)
  {
    isRed = '1';
    //invio il messaggio di accendere il rosso alla seconda scheda
    BTserial.write(isRed);
    //aspetto che il carrello AGV sia fuori dalla portata del sensore PIR
    delay(5000);
  }
}
```

# CODICE SCHEDA SEMAFORO/SLAVE - TEST COMUNICAZIONE BLUETOOTH

```
//CODICE DA INSTALLARE SULLA SCHEDA ARDUINO CON IL BLUETOOTH SLAVE (CON IL SEMAFORO) PER I TEST DI CONNESSIONE
//CON LA SCHEDA BLUETOOTH MASTER INSTALLATA SULLA SCHEDA ARDUINO CON IL SENSORE PIR
#include <SoftwareSerial.h>
SoftwareSerial BTSerial(12, 11); // RX | TX

void setup()
{
  Serial.begin(9600);
  Serial.println("LEGGI IL VALORE INVIATO ALLA SCHEDA SLAVE BLOOTOOH:");
  //le due schede devono comunicare con lo stesso baud di 9600
  BTSerial.begin(9600);
}

void loop()
{
  // leggo quello che la scheda bluetooth master ha inviato alla slave collegata alla presente scheda arduino
  // e lo scrivo su monitor seriale
  if (BTSerial.available())
    Serial.write(BTSerial.read());
}
```

# CODICE SCHEDA SEMAFORO/SLAVE – CODICE COMPLETO

```
//libreria per la connessione via bluetooth tra due schede bluetooth
#define pir 4
#include <SoftwareSerial.h>
//HC-06 è la scheda SLAVE qui gestita per ricevere un messaggio
SoftwareSerial BTserial(12, 11); // RX | TX
//il messaggio ricevuto è '1' per dire alla scheda di accendere il rosso
char isRed = '0';

int v1 = 2;
int g1 = 3;
int r1 = 4;
```

# CODICE SCHEDA SEMAFORO/SLAVE – CODICE COMPLETO

```
//dichiaro le procedure per accendere, rispettivamente, il semaforo verde sv(),
//quello giallo sg() e quello rosso sr()
void sv(int v, int g, int r);
void sg(int v, int g, int r);
void sr(int v, int g, int r);

/**inizializzazione
void setup() {
    //inizializzo la connessione tra le schede bluetooth in modalità seriale
    //usare lo stesso baud rate su master e slave
    BTserial.begin(9600);
}
}
```

# CODICE SCHEDA SEMAFORO/SLAVE – CODICE COMPLETO

```
/**progrmma di gestione del SEMAFORO e scheda BLUETOOTH SLAVE
void loop() {

    sv(v1,g1,r1); //semaforo verde
    //finchè non arriva nessun segnale dal bluettoth master
    //il semaforo resta verde
    while(isred == '0')
    {
        // leggo quello che la scheda blootooh master ha inviato alla slave collegata alla presente scheda arduino
        if (BTSerial.available())
            isRed=BTSerial.read();
    }

    sg(v1,g1,r1); //semaforo giallo
    delay(3000);

    sr(v1,g1,r1); //semaforo rosso
    //ASPETTO CHE IL CARRELLO AGV SIA PASSATO
    delay(5000);

}
```

# CODICE SCHEDA SEMAFORO/SLAVE – CODICE COMPLETO

```
//procedura per accendere il semaforo verde
//in input passo come parametri i pin dei tre colori verde, giallo e rosso del semaforo
void sv(int v, int g, int r){
    //accendo il verde
    digitalWrite(v,HIGH);
//spengo il giallo e il rosso
    digitalWrite(g,LOW);
    digitalWrite(r,LOW);
}

void sg(int v, int g, int r){
    digitalWrite(v,HIGH);
    digitalWrite(g,HIGH);
    digitalWrite(r,LOW);
}

void sr(int v, int g, int r){
    digitalWrite(v,LOW);
    digitalWrite(g,LOW);
    digitalWrite(r,HIGH);
}
```

# CONCLUSIONI

---

- ✘ Gestione del prototipo con:
  - + robot car kit per simulare il carrello AGV
  - + Tre led per simulare un semaforo
  - + Sensore PIR per rilevare il passaggio
  - + Scheda bluetooth master HC-05
  - + Scheda bluetooth slave HC-06