

Lezione 2 Macchina Bluetooth



Punti di questa sezione

E' molto importante e divertente controllare la tua macchina via wireless in un certo spazio quando di impara ad usare Arduino, cosi' in questa lezione, cosi' in questa lezione ti insegneremo a controllare la macchina via Bluetooth.

Sezioni di apprendimento:

- Apprendere come usare il modulo Bluetooth CC41-A e la APP Bluetooth
- Apprendere come controllare il veicolo via Bluetooth
- Scrivere I programmi che implementino questa funzione

Componenti necessari:

- Un veicolo (equipaggiato con la batteria)
- Un cavo USB
- Un modulo Bluetooth CC41-A
- Un iPhone o tablet

I . Modulo Bluetooth



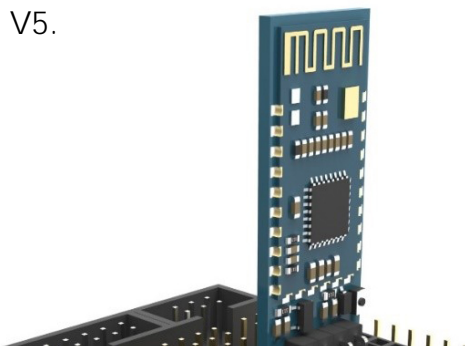
T

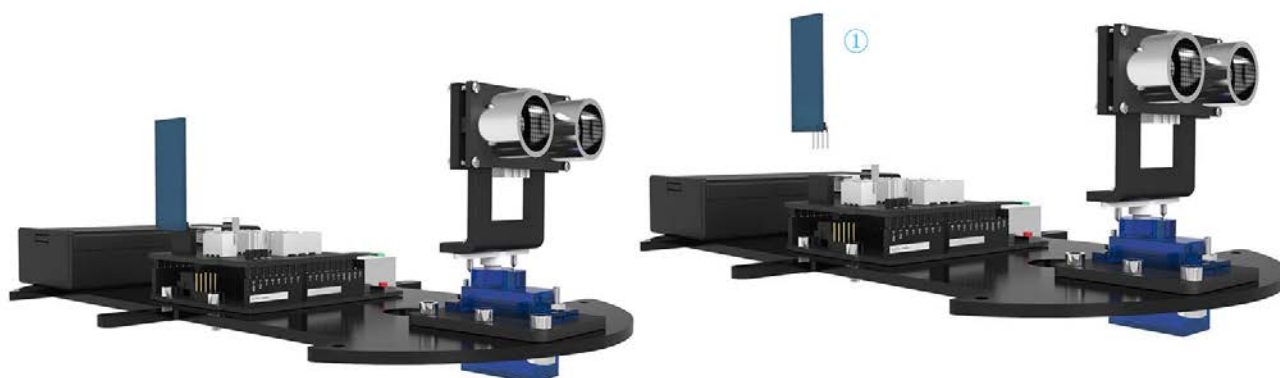
Descrizione del modulo Bluetooth CC41-A :

1. Adotta un chip Bluetooth largamente usato della CSR, con un protocollo standard Bluetooth V2.0
2. Il voltaggio analogico di lavoro della porta seriale e' di 3.3V
3. L'utente puo' impostare la velocita' a 1200, 2400, 4800, 9600, 19200, 38400, 57600,115200
4. Le dimensioni del modulo sono 28mm x 15 mm x2.35mm
5. La corrente elettrica di lavoro e': 40MA
6. La corrente elettrica a riposo e': meno di 1MA
7. Puo' essere usato per sistemi di navigazione GPS, sistemi di lettura dei gas idroelettrici, sistemi di controllo di estrazione industriali
8. Puo' essere connesso a portatili Bluetooth, computer con adattatori Bluetooth , PDA, ecc.

Questo e' il diagramma schematico del modulo Bluetooth connesso alla scheda di controllo UNO:

Nell'esperimento, lo conatteremo alla scheda UNO attraverso la scheda di espansione V5.

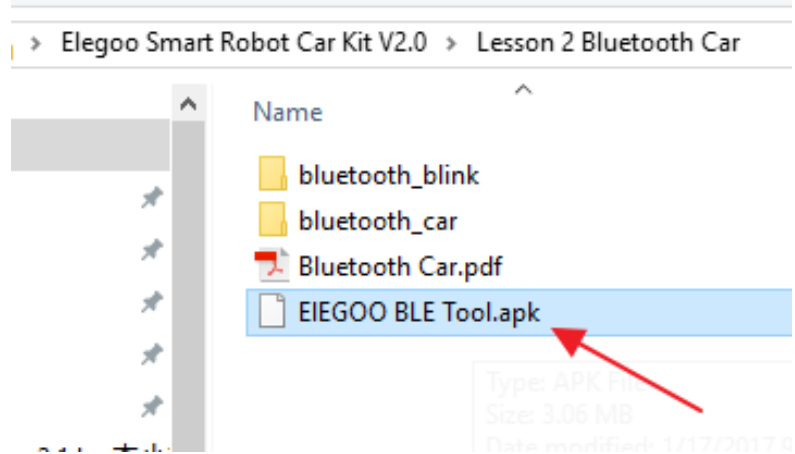




① bluetooth module

II . La APP Bluetooth

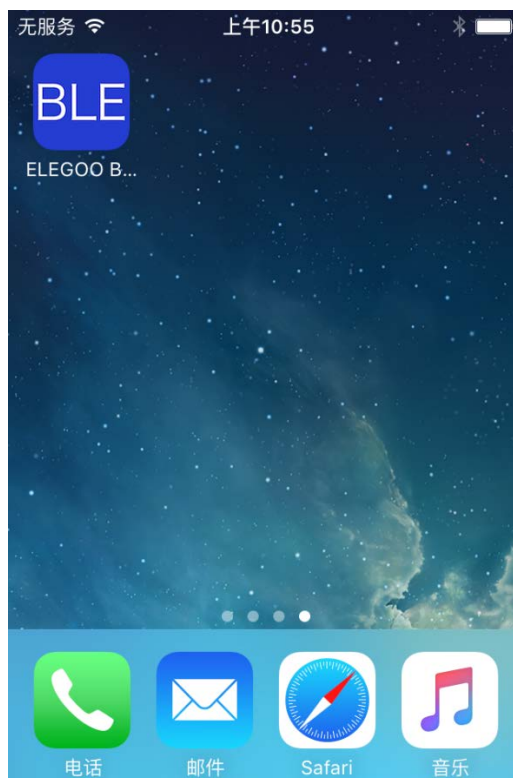
Per I sistemi Android, copiate il file APK sul dispositivo Android ed installatelo.



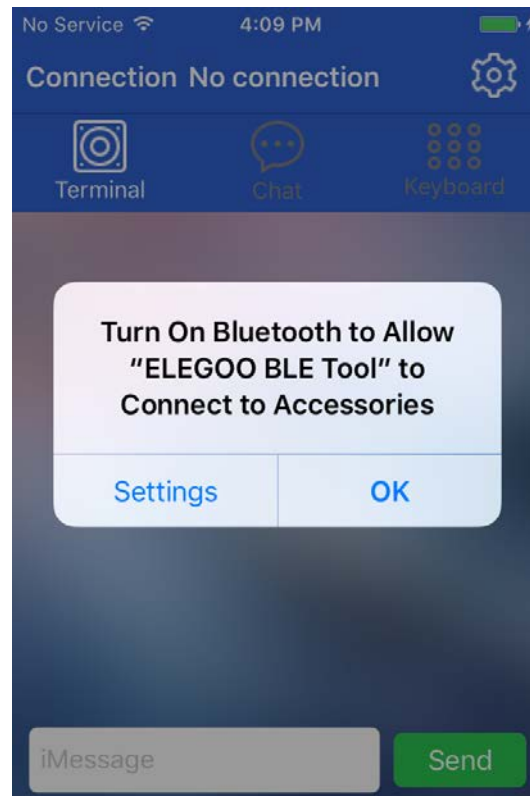
Sotto, prendento un telefono Apple ad esempio:

Cercate nell' Apple Store "ELEGOO BLE Tool".

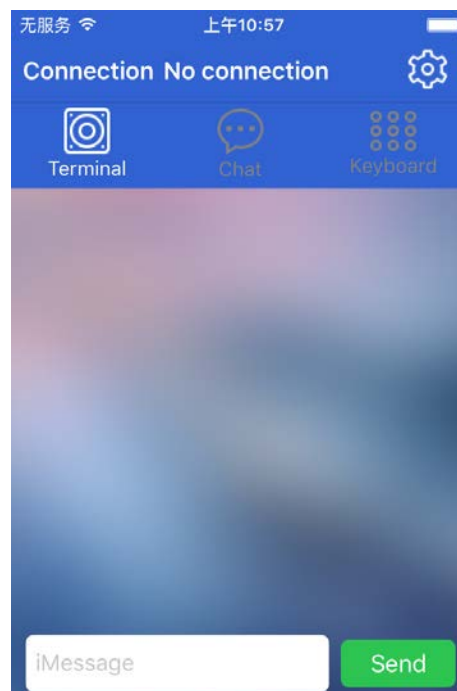
Installate la APP e lanciatela sul telefono.



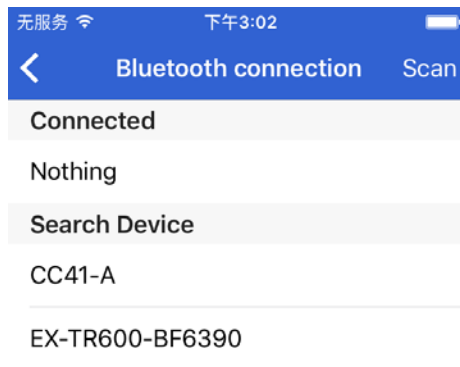
Vi mostrera' un avviso per avvisarvi di attivare il Bluetooth quando la lanciate.



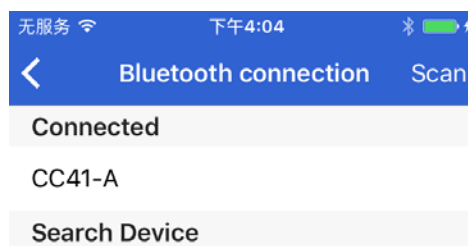
Cliccate Connetti.



A questo punto il vostro telefono cercherà i dispositivi Bluetooth nelle vicinanze.

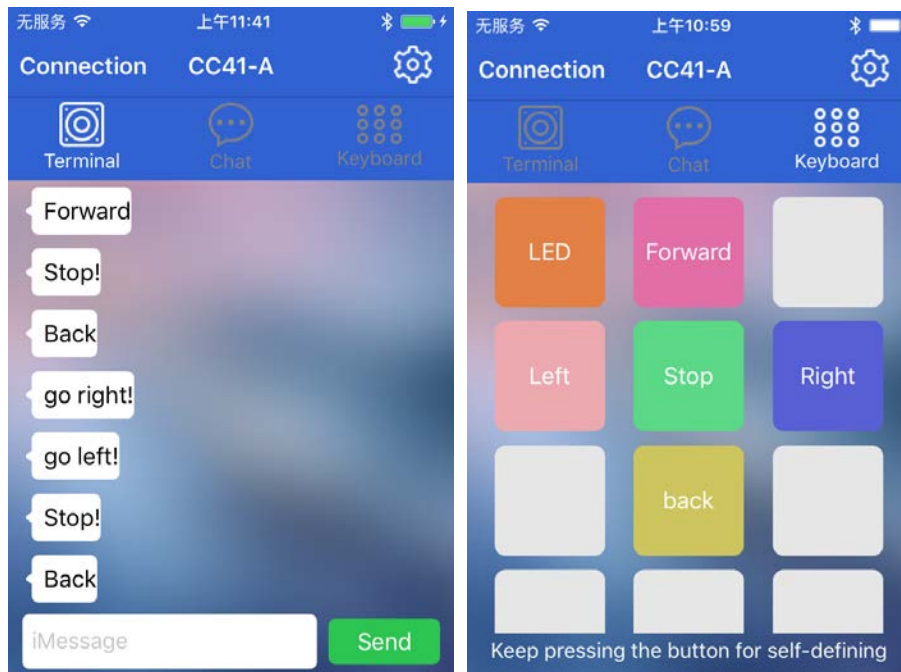


Cliccate su CC41-A , e quando la connessione sara' avvenuta sullo schermo leggerete: Periferica Connessa

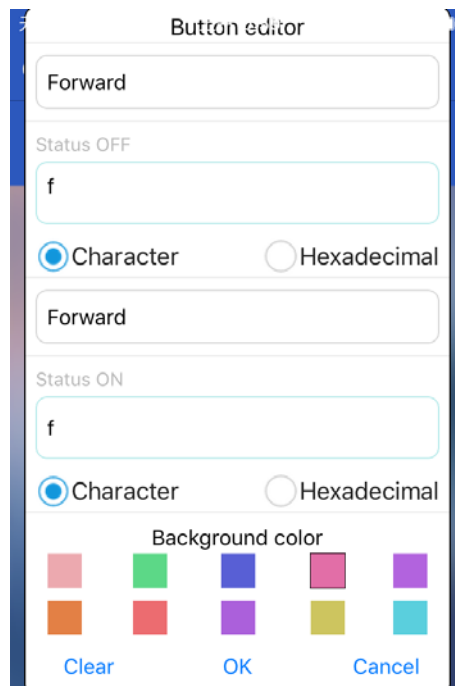


Device Connected

L'interfaccia della app apparira' come mostrato sotto a sinistra, potremo allora scorrere lo schermo a sinistra con le dita ed ottenere gli schemi di base come nell'immagine sotto a destra:



In fine imposteremo la funzione di ogni pulsante, prendiamo 'vai-avanti' come esempio, come potete vedere sotto, I restanti pulsanti potranno essere impostati con lo stesso procedimento.

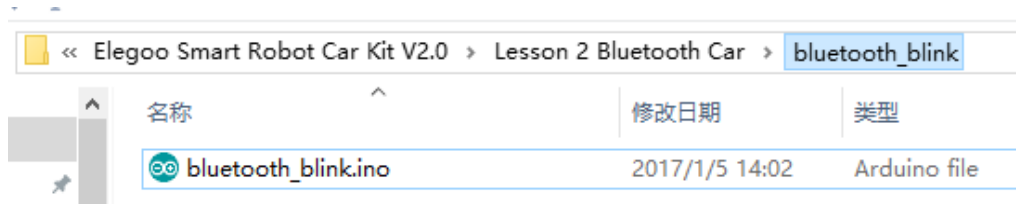


III. Test

Testiamo il programma con questo codice:

```
int LED=13;//Define 13 pin of LED
volatile int state = LOW;//The initial state of function is defined as a low level
char getstr;    //Defines function that receives Bluetooth character
void setup()
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
}
/*Control LED sub function*/
void stateChange()
{
  state = !state;
  digitalWrite(LED, state);
}
void loop() {
  getstr=Serial.read();//Bluetooth serial port receives data in function
  if(getstr=='A')
  {
    stateChange();
  }
}
```

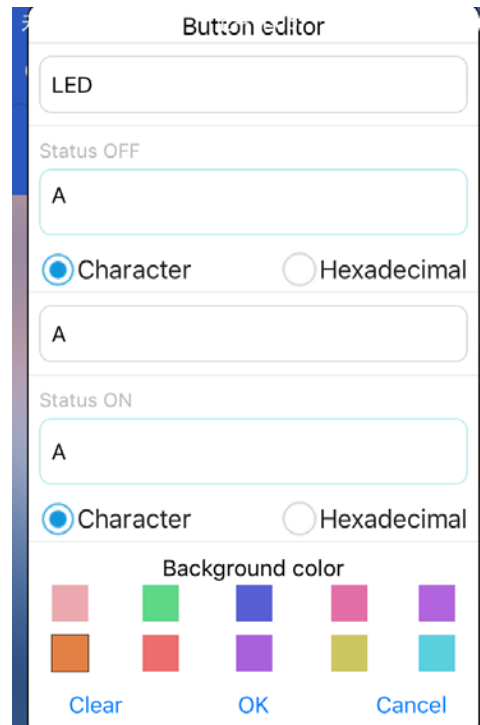
Apriamo il file : \bluetooth_blink\bluetooth_blink.ino



Eseguiamo l'Upload del programma sulla scheda UNO, disconnettiamola dal computer, e accendiamo l'alimentazione della macchina. (Avviso: il modulo Bluetooth va rimosso quando fate l'upload del programma, o questo non adra' a buon fine.)

Apriamo la APP

Dopo aver connesso il telefono alla macchina con il Bluetooth, imposteremo i dati come mostrato sotto:



Dopo il set-up, premete questo pulsante. Potrete vedere che la luce sulla scheda uno cambierà con il comando.

il codice

```
Serial.begin(9600);
```

Lo scopo di questo blocco di codice è impostare la velocità della scheda controller UNO a 9600 e aprire la porta seriale. In questo modo, potranno comunicare tra di loro, perché la velocità originale del modulo Bluetooth è 9600.

```
getstr=Serial.read();//The Bluetooth serial port to receive the data in the function
```

```
if(getstr=='A'){
```

```
stateChange();
```

Questa funzione viene eseguita continuamente con un ciclo. Prima legge i dati dalla porta seriale, li controlla, e se corrispondono a certe condizioni, eseguono la corrispondente sotto-funzione. Per esempio, se legge la lettera A dalla porta seriale,

eseguirà la sotto-funzione addetta ad accendere e spegnere la luce LED .

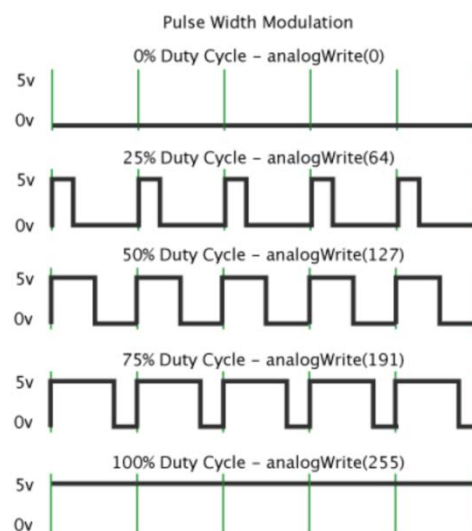
IV. Creare la macchina Bluetooth

Quando la macchina gira a destra o a sinistra, non e' necessario impostare la velocita' troppo elevata. In altri casi e' necessario controllare la velocita' della macchina. Bma come farlo?

La risposta e' PWM.

PWM e' la abbreviazione di "Pulse Width Modulation" (impulso con modulazione), viene chiamato in breve, modulazione dell'impulso, e' una tecnologia efficace per controllare dei circuiti analogici con una uscita digitale del microprocessore, la macchina cambia la velocita' del motore alterando i cicli intermittenti di un onda quadra. In altre parole, connette ed interrompe il circuito tra le due parti del motore costantemente, interrompendo il lavoro del motore, ma il motor non si fermere' a causa della alta velocita'. Cosi' possiamo controllare la velocita' della macchina controllando specifici valori di accensione e spegnimento. La velocita' della macchina sara' massima quando il circuito verra' mantenuto costantemente attivo. La velocita' della macchina sara' la minima, quando il circuito verra' mantenuto spento.

La velocita' della macchina sara' media, se il circuito sara' attivo per la meta' del tempo. PWM e' una tecnologia che permette di ottenere valori analogici attraverso metodi digitali. Un controllo digitale crea un onda quadra, i segnali ad onda quadra hanno solamente due valori: acceso o spento (corrispondono as alto-basso dei pin digitali). Simulare la variazione di voltaggio tra 0 e 5V controllando specifici valori di attivazione e disattivazione. Il tempo di attivazione (Alto livello , scolasticamente) viene chiamato ampiezza d'impulso, infatti il PWM viene anche chiamato modulazione dell'ampiezza d'impulso. Studiamo il PWM guardando le cinque onde quadre qui sotto.



Le linee verticali verdi sopra, rappresentano la frequenza dell'onda quadra. Il valore scritto in ogni `analogWrite(valore)` corrisponde alla percentuale, la percentuale viene chiamata anche Duty Cycle, e si riferisce alla percentuale ottenuta da valori specifici tra la durata del livello attivo (high) ed inattivo (low) del segnale in un ciclo. In figura, dall'alto al basso, nella prima onda quadra, il duty cycle e' 0%, e il valore corrispondente e' 0. la corrente in uscita dal circuito e' la minima, il motore e' fermo. Piu' e' lungo il tempo di attivazione, maggiore sara' la corrente che arrivera' al motore e quindi anche la velocita' sara' superiore. L'ultimo duty cycle e' 100%, e il valore corrispondente e' 255, il motore ruota alla massima velocita'. 50% e' la velocita' di rotazione iponastica meda, 25% e' relativamente piu' lenta, ma non puo' farlo partire (la corrente di circuito necessaria a far partire il motore e' relativamente grande a causa dell'attrito statico). Il PWM viene usato spesso per regolare la luminosita' dei LED e la velocita' di rotazione dei motori, e' semplice cosi' controllare la velocita' delle ruote controllate dal motore. Il vantaggio nell'usare il PWM diventa evidente giocando con le macchine gestite da Arduino .

`analogWrite(pin,value);`

`analogWrite()` viene usato per scrivere un valore analogico da 0 a 255 per le porte PWM . Notate che , `analogWrite()` viene usato unicamente sui pin digitali con funzione di PWM. Pi in con funzione di PWM in UNO sono solo i pin digitali 3, 5, 6, 9, 10, 11.

La velocita' della nostra macchina verra' controllata connettendo i pin pin5 e pin10 di ENA e ENB. Il programma qui sotto ha impostata la funzione digitale `int ABS=135;` La velocita' viene controllata dal programma qui sotto, in questo modo potrete controllare la velocita' da soli.

`analogWrite(ENA,ABS);`

`analogWrite(ENB,ABS);`

Dopo aver appreso le conoscenze di base, eseguiremo l'upload del programma come indicato qui sotto; aprirete il file `bluetooth_car\bluetooth_car.ino`.

`int LED=13;`

`volatile int state = LOW;`

`char getstr;`

`int in1=9;`

```
int in2=8;
int in3=7;
int in4=6;
int ENA=10;
int ENB=5;
int ABS=135;
void _mForward()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,LOW);
    digitalWrite(in4,HIGH);
    Serial.println("go forward!");
}
void _mBack()
{
    digitalWrite(ENA,HIGH);
    digitalWrite(ENB,HIGH);
    digitalWrite(in1,HIGH);
    digitalWrite(in2,LOW);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("go back!");
}
void _mleft()
{
    analogWrite(ENA,ABS);
    analogWrite(ENB,ABS);
    digitalWrite(in1,LOW);
    digitalWrite(in2,HIGH);
    digitalWrite(in3,HIGH);
    digitalWrite(in4,LOW);
    Serial.println("go left!");
}
```

```
void _mright()
{
  analogWrite(ENA,ABS);
  analogWrite(ENB,ABS);
  digitalWrite(in1,HIGH);
  digitalWrite(in2,LOW);
  digitalWrite(in3,LOW);
  digitalWrite(in4,HIGH);
  Serial.println("go right!");
}

void _mStop()
{
  digitalWrite(ENA,LOW);
  digitalWrite(ENB,LOW);
  Serial.println("Stop!");
}

void stateChange()
{
  state = !state;
  digitalWrite(LED, state);
}

void setup()
{
  pinMode(LED, OUTPUT);
  Serial.begin(9600);
  pinMode(in1,OUTPUT);
  pinMode(in2,OUTPUT);
  pinMode(in3,OUTPUT);
  pinMode(in4,OUTPUT);
  pinMode(ENA,OUTPUT);
  pinMode(ENB,OUTPUT);
  _mStop();
}

void loop()
{
  getstr=Serial.read();
```

```
if(getstr=='f')
{
  _mForward();
}
else if(getstr=='b')
{
  _mBack();
  delay(200);
}
else if(getstr=='l')
{
  _mleft();
  delay(200);
}
else if(getstr=='r')
{
  _mright();
  delay(200);
}
else if(getstr=='s')
{
  _mStop();
}
else if(getstr=='A')
{
  stateChange();
}
}
```

Fate l'Upload del programma alla scheda di controllo UNO , accendete il veicolo e poggiatelo a terra.

Aprirete la APP mobile, e impostate i parametri come qui di seguito.

Button editor

Forward

Status OFF

f

☒ Character ☐ Hexadecimal

Forward

Status ON

f

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

Button editor

back

Status OFF

b

☒ Character ☐ Hexadecimal

back

Status ON

b

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

Button editor

Left

Status OFF

l

☒ Character ☐ Hexadecimal

Left

Status ON

l

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

Button editor

Right

Status OFF

r

☒ Character ☐ Hexadecimal

Right

Status ON

r

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

Button editor

Stop

Status OFF

s

☒ Character ☐ Hexadecimal

Stop

Status ON

s

☒ Character ☐ Hexadecimal

Background color

ClearOKCancel

无服务 上午10:59

Connection CC41-A

Terminal Chat Keyboard

LED Forward

Left Stop Right

back

Keep pressing the button for self-defining