```
#include <SoftwareSerial.h>
#include <Servo.h>

#define DELAY_BLINK 450 //ms

SoftwareSerial BTserial(3, 2); // RX | TX

#define TURN_LEFT_SIGNAL 7
#define TURN_RIGHT_SIGNAL 8
#define FAR_LIGHT 9
#define CAR_HORN 10

#define TURN_PIN 5
#define GO_PIN 6
#define MIN_TURN 850
#define MAX_TURN 2150
#define STOP_SPEED 1500
#define MAX_SPEED_GO 2500
#define MAX_SPEED_BACK 500
Servo turn;
Servo sgo;

char command;
String string;
int svangle = 0;
int slideBarValue = 50;
int index = 0;
String aCmd;

int speeds = STOP_SPEED;
int gear = 0;
```

```
int turn_blink_state = LOW;
int turn_status_cmd = 0;
long previousMillis = 0;

void carGo(int st){
  sgo.writeMicroseconds(st);
  delay(10);
}

void setup()
{
  //Serial.begin( 9600 );//115200
  BTserial.begin( 38400 );
  pinMode(LED_BUILTIN, OUTPUT);
  turn.attach(TURN_PIN);
  svangle = map(slideBarValue, 0, 100, MIN_TURN, MAX_TURN);
  turn.writeMicroseconds(svangle);

  sgo.attach(GO_PIN);
  sgo.writeMicroseconds(speeds);

  pinMode(TURN_LEFT_SIGNAL, OUTPUT);
  pinMode(TURN_RIGHT_SIGNAL, OUTPUT);
  pinMode(FAR_LIGHT, OUTPUT);
  pinMode(CAR_HORN, OUTPUT);

  light_off();
  turn_signal_off();

  //Serial.println("Setup done!");
}
```

```
void car_horn_on(){
  tone(CAR_HORN, 410);
}

void car_horn_off(){
  noTone(CAR_HORN);
}

void blinkLED(){
  if ( turn_blink_state == LOW ){
    turn_blink_state = HIGH;
  } else {
    turn_blink_state = LOW;
  }
  if( turn_status_cmd == 1 ){
    // turn left
    digitalWrite(TURN_LEFT_SIGNAL, turn_blink_state);
    digitalWrite(TURN_RIGHT_SIGNAL, HIGH);
  } else if( turn_status_cmd == 2 ){
    // turn right
    digitalWrite(TURN_LEFT_SIGNAL, HIGH);
    digitalWrite(TURN_RIGHT_SIGNAL, turn_blink_state);
  } else if( turn_status_cmd == 3 ){
    digitalWrite(TURN_LEFT_SIGNAL, turn_blink_state);
    digitalWrite(TURN_RIGHT_SIGNAL, turn_blink_state);
  } else {
    turn_signal_off();
  }
}
```

```
void turn_signal_off(){
  digitalWrite(TURN_LEFT_SIGNAL, HIGH);
  digitalWrite(TURN_RIGHT_SIGNAL, HIGH);
}

void far_light(){
  digitalWrite(FAR_LIGHT, LOW);
}

void light_off(){
  digitalWrite(FAR_LIGHT, HIGH);
}

void ledOn()
{
  digitalWrite(LED_BUILTIN, HIGH);
  delay(10);
}

void ledOff()
{
  digitalWrite(LED_BUILTIN, LOW);
  delay(10);
}

void writeString(String stringData) { // Used to serially push out a String with Serial.write()
  for (int i = 0; i < stringData.length(); i++)
  {
    BTserial.write(stringData[i]);   // Push each char 1 by 1 on each loop pass
  }
}
```

```
void sendAck(String toSend){
  char payload[toSend.length()+1];
  toSend.toCharArray(payload, sizeof(payload));
  BTserial.write((uint8_t *)payload,sizeof(payload));
}

void loop()
{
 // for blinking turn signal
 unsigned long currentMillis = millis();
 if( currentMillis - previousMillis >= DELAY_BLINK ){
   previousMillis = currentMillis;
   blinkLED();
 }
 // for receiving bluetooth data
 string = "";
 while(BTserial.available() > 0)
 {
   command = ((byte)BTserial.read());
   if(command == ':')
   {
     break;
   }
   else
   {
     string += command;
   }
   delay(1);
 }
 //if(string != "")  Serial.println(string);
```

```
while( string.length() >= 3 ){
  aCmd = string.substring(0, 3);
  string = string.substring(3);
  //Serial.println(" " + aCmd);

  index = aCmd.lastIndexOf("T");
  if( aCmd == "GOO"  ){
    // Move the car
    carGo(MAX_SPEED_GO);
  } else if( aCmd == "STG" ){
    carGo(STOP_SPEED);
    // Stop the car
  } else if( aCmd == "BAC" ){
    // Move the car back
    carGo(MAX_SPEED_BACK);
  } else if( aCmd == "STB" ){
    // Stop the car
    carGo(STOP_SPEED);
  } else if( index == 0 ){
    // Turn left/right: cmd = "T<value from 0 to 100>"
      slideBarValue = aCmd.substring(index+1).toInt();
      //Serial.println(slideBarValue );
      if( slideBarValue > 0 ){
        //turn.attach(TURN_PIN);
        svangle = map(slideBarValue, 0, 100, MIN_TURN, MAX_TURN);
        turn.writeMicroseconds(svangle);
      }
  } else if ( aCmd.lastIndexOf("S") == 0 ){
    speeds = aCmd.substring(1).toInt();
    if( speeds > 0 ){
```

```
        speeds -= 15;
        if( gear == 3 ){
          sgo.writeMicroseconds( map(speeds, 0, 100, STOP_SPEED, MAX_SPEED_GO) );
        } else if( gear == 1 ){
          sgo.writeMicroseconds( map(speeds, 0, 100, STOP_SPEED, MAX_SPEED_BACK) );
        }
        delay(10);
      }
    } else if ( aCmd.lastIndexOf("G") == 0 ){
      gear = aCmd.substring(1).toInt();
    } else if (aCmd.lastIndexOf("R") == 0 ){
      if( aCmd == "REF" ) turn_status_cmd = 1;
      else if( aCmd == "RHT" ) turn_status_cmd = 2;
      else if( aCmd == "RAA" ) turn_status_cmd = 3;
      else turn_status_cmd = 0;
    } else if (aCmd.lastIndexOf("L") == 0 ){
      if( aCmd == "LFA" ) far_light();
      //else if( aCmd == "LNE" ) near_light();
      else light_off();
    } else if (aCmd.lastIndexOf("H") == 0 ){
      if( aCmd == "HON" ) car_horn_on();
      else car_horn_off();
    }
  }
}
```