Princípios de comunicação bluetooth

Prof. Me. Hélio Esperidião

A Tecnologia Bluetooth

- Desenvolvida originalmente pela Ericsson e gerenciada por um grupo de empresas - o Bluetooth SIG (Special Interest Group)
- Substituição de cabos de conexão
- Trabalha na banda de 2.4 GHz utilizando FHSS com 1600 saltos/s
- Transferências da ordem de centenas de Kbps
- Cada aparelho possui um ID único para facilitar identificação

Requisitos da Tecnologia

- Baixo custo
  - Para poder ser embutido em outros dispositivos sem modificar sua faixa de preço
- Baixo consumo de energia
  - Utilização voltada para dispositivos móveis
- Tamanho Reduzido
- Pequeno alcance

Padrão IEEE 802.15.1

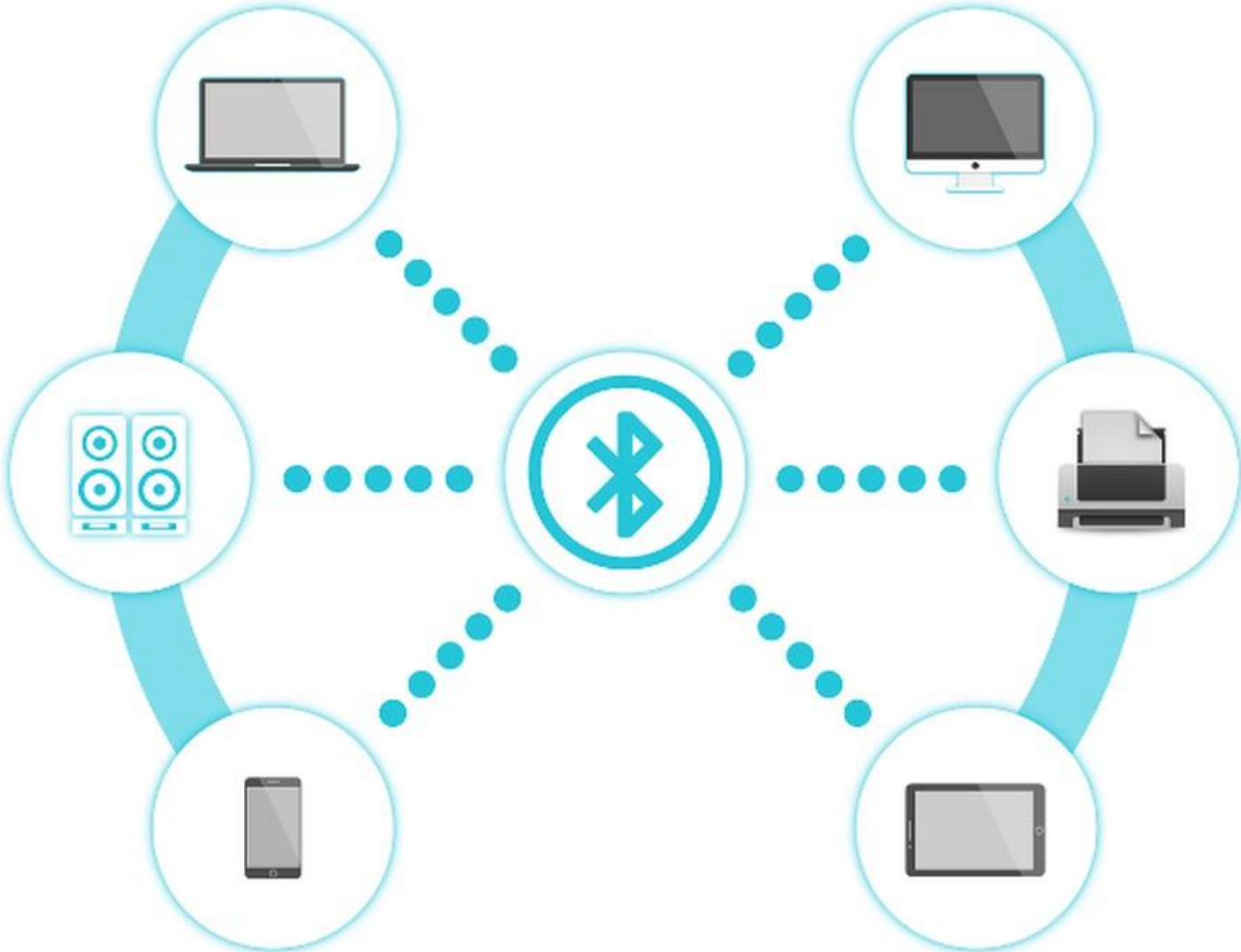
- Objetivos
  - Baixo custo
  - Baixo consumo energético
  - Leve
  - Fácil uso
  - Confiável e tolerante a falhas

Padrão IEEE 802.15.1

- Frequência —> 2,4 GHz
- Resiliente a falhas
- Taxa de transmissão
  - 1 Mbs
  - Distância
  - média 10 metros
  - máxima 100 metros

Bluetooth

- O que é Bluetooth?
  - Interface de rádio universal
  - Pequena distância (10 m)
  - Conexão entre dispositivos eletrônicos portáteis
  - Redes Ad Hoc(não há um nó central)
  - Elimina necessidade cabos em modems, PDAs
  - computadores, impressoras, projetores, etc.
  - Baixa pôtência, custo e complexidade



Bluetooth

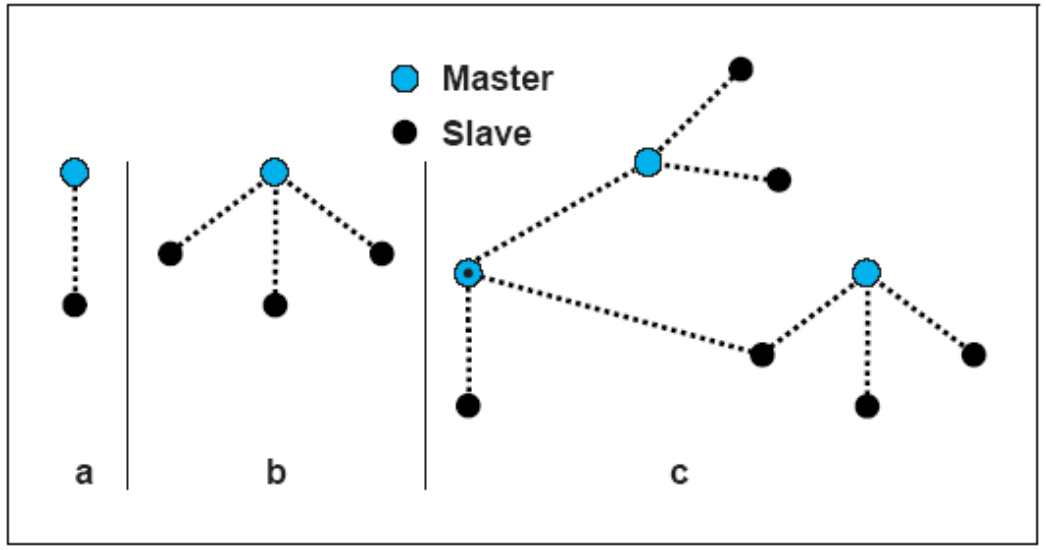
- Conectividade em sistemas sem fio
  - Em sistemas de rádio convencionais, o móvel se conecta a uma estação base que gerencia a comunicação entre este móvel e outros terminais.
- O acesso ao canal, alocação de canal, controle de tráfego e minimização de interferência são controlados pela estação base.
  - Exemplos: GSM. D-AMPS, IS-95, WLAN, etc

Bluetooth

- Conectividade em sistemas AD Hoc
  - Em sistemas AD Hoc não existe diferenciação entre terminal e estação base.
  - Comunicação ponto-a-ponto
  - Não existe central de controle para que as unidades possam fazer as conexões ou para dar suporte a estas conexões
  - Várias conexões AD Hoc podem compartilhar o meio sem necessidade de uma estação de controle

Topologia das redes

- Formação de redes ad-hoc formando piconets
  - Cada piconet tem sempre um dispositivo mestre e ate 7 escravos



Topologia das redes

- Toda a comunicação é feita entre o mestre e um escravo
- Quando *duas ou mais piconets compartilham algum dispositivos*, temos uma *scatternet*
- Esses dispositivos atuam como roteadores entre as duas redes

Bluetooth

- Espectro de Frequência
  - O sistema deve operar em qualquer parte do mundo e a banda de frequência deve ser aberta ao público sem a necessidade de licenças
- A única banda de frequência que satisfaz estes requisitos é a 2,45 GHz - Industrial-Scientific medical (ISM) band
- 2400 MHz à 2483,5 MHz nos EUA e na Europa (apenas parte desta banda está disponível na França e Espanha)
- 2471 MHz à 2497 MHz no Japão

Bluetooth

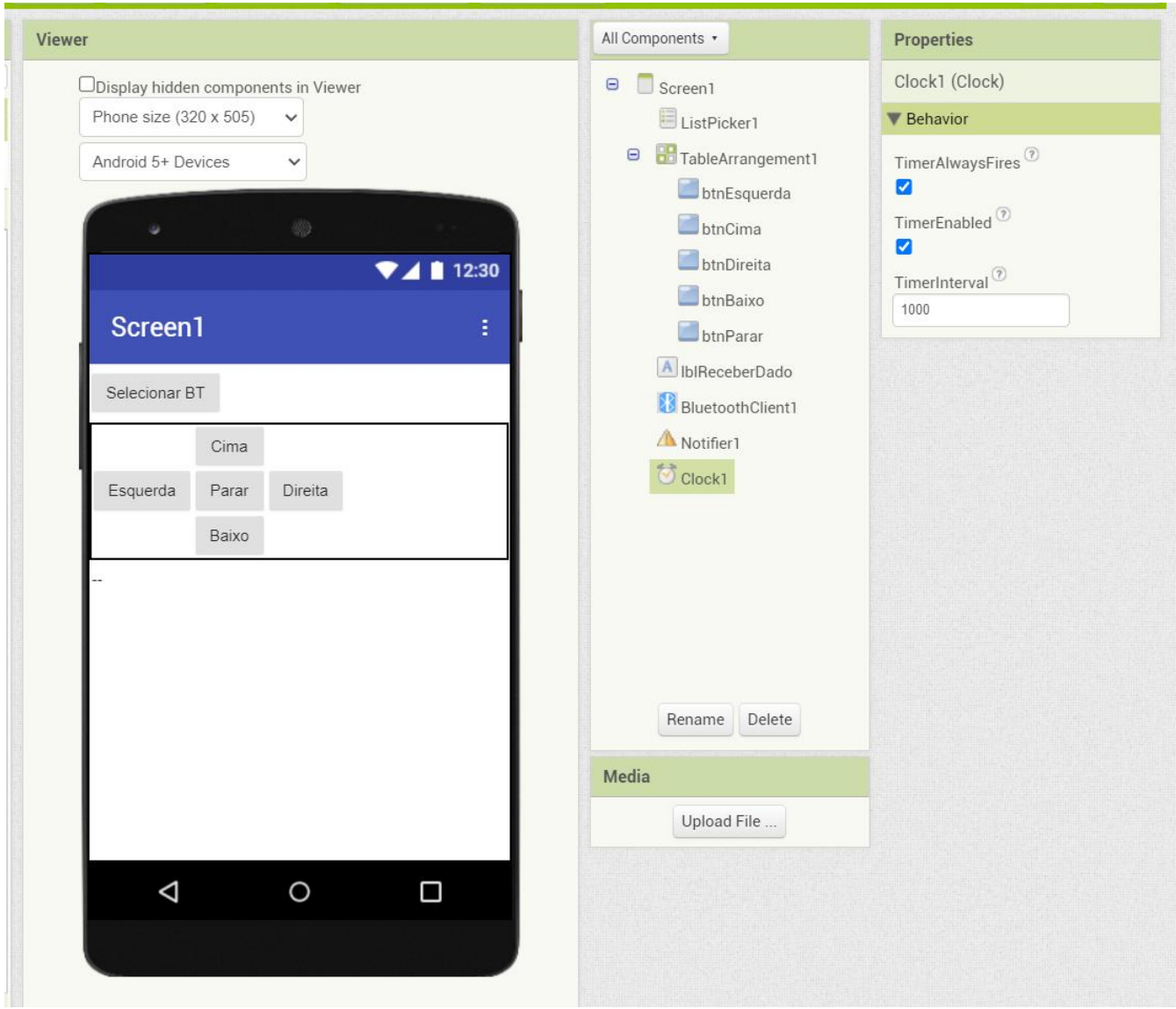
- Bluetooth 4.2
  - Frequência - 2.4 GHz
  - Alcance - 30m
  - Velocidade - 1Mbps
- Bluetooth 5.0
  - Consumo de Energia
  - Velocidade
  - Expansão na conexão - Mesh (IoT)

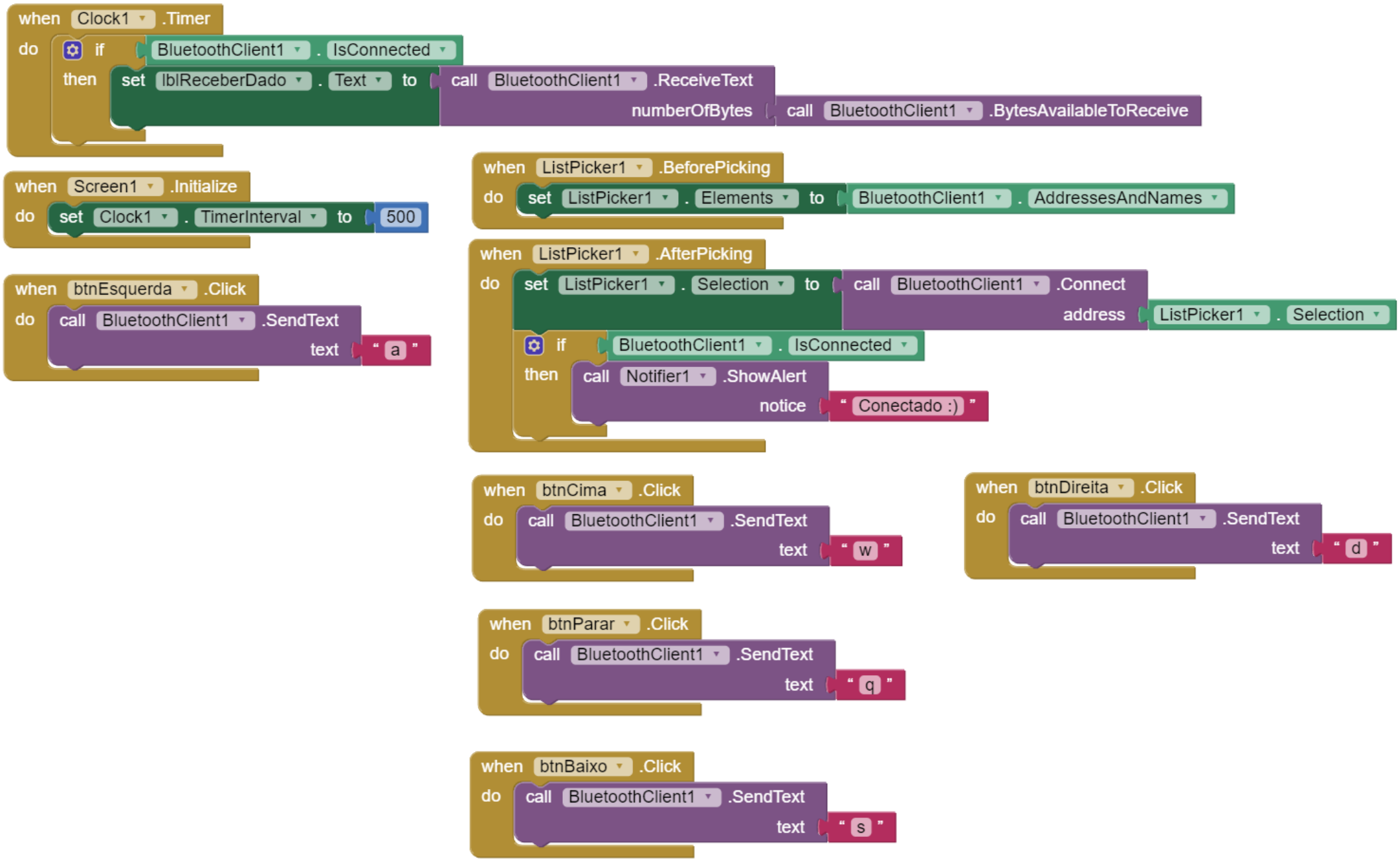
Bluetooth

```
#include "BluetoothSerial.h"
BluetoothSerial SerialBT;
#define BT_NAME "UNIVAP"
void setup()
{
  Serial.begin(115200); // Inicializa o monitor serial

  SerialBT.begin(BT_NAME); // Nome do dispositivo Bluetooth
  Serial.println("ESP32 Bluetooth Iniciado. Aguardando conexão...");
}
```

```
void loop()
{
  if (SerialBT.available())
  {                                                  // Verifica se há dados recebidos via Bluetooth
    char received = SerialBT.read();                 // Lê o primeiro caractere recebido
    Serial.println("Recebido: " + String(received));
    SerialBT.println("ESP32 Recebeu: d “ + String(received));
   }
}
```


Viewer
Display hidden components in Viewer
Phone size (320 x 505)
Android 5+ Devices
12:30
Screen1
Selecionar BT
Cima
Esquerda
Parar
Direita
Baixo
--
All Components
Screen1
ListPicker1
TableArrangement1
btnEsquerda
btnCima
btnDireita
btnBaixo
btnParar
lblReceberDado
BluetoothClient1
Notifier1
Clock1
Rename
Delete
Media
Upload File ...
Properties
Clock1 (Clock)
Behavior
TimerAlwaysFires
TimerEnabled
TimerInterval
1000


when Clock1 .Timer
do if BluetoothClient1 . IsConnected
then set lblReceberDado . Text to call BluetoothClient1 .ReceiveText
numberOfBytes call BluetoothClient1 .BytesAvailableToReceive
when Screen1 .Initialize
do set Clock1 . TimerInterval to 500
when btnEsquerda .Click
do call BluetoothClient1 .SendText
text " a "
when ListPicker1 .BeforePicking
do set ListPicker1 . Elements to BluetoothClient1 . AddressesAndNames
when ListPicker1 .AfterPicking
do set ListPicker1 . Selection to call BluetoothClient1 .Connect
address ListPicker1 . Selection
if BluetoothClient1 . IsConnected
then call Notifier1 .ShowAlert
notice " Conectado :) "
when btnCima .Click
do call BluetoothClient1 .SendText
text " w "
when btnDireita .Click
do call BluetoothClient1 .SendText
text " d "
when btnParar .Click
do call BluetoothClient1 .SendText
text " q "
when btnBaixo .Click
do call BluetoothClient1 .SendText
text " s "

```
#include <Arduino.h>
#include "BluetoothSerial.h"

BluetoothSerial SerialBT;

#define BT_NAME "UNIVAP" // Nome do dispositivo Bluetooth

/*
  LIGAÇÃO DO MOTOR
  MOTOR A         IN1       IN2
  direto           1         0
  reverso          0         1
  parar            0         0
  parar            1         1
*/

#define IN1_MOTOR1 27
#define IN2_MOTOR1 26

#define IN3_MOTOR2 25
#define IN4_MOTOR2 33
```

```
void moverFrente(){
  digitalWrite(IN1_MOTOR1, 1);
  digitalWrite(IN2_MOTOR1, 0);

  digitalWrite(IN3_MOTOR2, 1);
  digitalWrite(IN4_MOTOR2, 0);
  SerialBT.println("ESP32 Recebeu: w ");
  Serial.println("Movendo para frente");
}
void moverTras(){
  digitalWrite(IN1_MOTOR1, 0);
  digitalWrite(IN2_MOTOR1, 1);

  digitalWrite(IN3_MOTOR2, 0);
  digitalWrite(IN4_MOTOR2, 1);
  SerialBT.println("ESP32 Recebeu: s ");
  Serial.println("Movendo para trás");
}
```

```
void moverEsquerda()
{
  digitalWrite(IN1_MOTOR1, LOW);
  digitalWrite(IN2_MOTOR1, HIGH);

  digitalWrite(IN3_MOTOR2, HIGH);
  digitalWrite(IN4_MOTOR2, LOW);
  SerialBT.println("ESP32 Recebeu: a ");
  Serial.println("Movendo para a esquerda");
}

void moverDireita()
{
  digitalWrite(IN1_MOTOR1, HIGH);
  digitalWrite(IN2_MOTOR1, LOW);

  digitalWrite(IN3_MOTOR2, LOW);
  digitalWrite(IN4_MOTOR2, HIGH);
  SerialBT.println("ESP32 Recebeu: d ");
  Serial.println("Movendo para a direita");
}
```

```
void parar()
{
  digitalWrite(IN1_MOTOR1, LOW);
  digitalWrite(IN2_MOTOR1, LOW);

  digitalWrite(IN3_MOTOR2, LOW);
  digitalWrite(IN4_MOTOR2, LOW);
  Serial.println("Parando os motores");
}

void setup(){
  Serial.begin(115200); // Inicializa o monitor serial
  pinMode(IN1_MOTOR1, OUTPUT);
  pinMode(IN2_MOTOR1, OUTPUT);

  pinMode(IN3_MOTOR2, OUTPUT);
  pinMode(IN4_MOTOR2, OUTPUT);

  SerialBT.begin(BT_NAME); // Nome do dispositivo Bluetooth
  Serial.println("ESP32 Bluetooth Iniciado. Aguardando conexão...");
}
```

```
void parar()
{
  digitalWrite(IN1_MOTOR1, LOW);
  digitalWrite(IN2_MOTOR1, LOW);

  digitalWrite(IN3_MOTOR2, LOW);
  digitalWrite(IN4_MOTOR2, LOW);
  Serial.println("Parando os motores");
}

void setup(){
  Serial.begin(115200); // Inicializa o monitor serial
  pinMode(IN1_MOTOR1, OUTPUT);
  pinMode(IN2_MOTOR1, OUTPUT);

  pinMode(IN3_MOTOR2, OUTPUT);
  pinMode(IN4_MOTOR2, OUTPUT);

  SerialBT.begin(BT_NAME); // Nome do dispositivo Bluetooth
  Serial.println("ESP32 Bluetooth Iniciado. Aguardando conexão...");
}
```

```
void loop(){
  if (SerialBT.available())  {   / Verifica se há dados recebidos via Bluetooth
    char received = SerialBT.read();                  // Lê o primeiro caractere recebido
    Serial.println("Recebido: " + String(received)); // Exibe no monitor serial
    switch (received)   {// Interpreta os comandos recebidos
    case 'w':
      moverFrente();
      break;
    case 's':
      moverTras();
      break;
    case 'a':
      moverEsquerda();
      break;
    case 'd':
      moverDireita();
      break;
    case 'q':
      parar();
      break;
    default:
      Serial.println("Comando inválido");
      break;
    }
  }
}
```