Código-base:

```
#include <XboxSeriesXControllerESP32_asukiaaa.hpp>

// Required to replace with your xbox address
// XboxSeriesXControllerESP32_asukiaaa::Core
// xboxController("44:16:22:5e:b2:d4");

// any xbox controller
XboxSeriesXControllerESP32_asukiaaa::Core xboxController;

void setup() {
  Serial.begin(115200);
  Serial.println("Starting NimBLE Client");
  xboxController.begin();
}

void loop() {
  xboxController.onLoop();
  if (xboxController.isConnected()) {
    if (xboxController.isWaitingForFirstNotification()) {
      Serial.println("waiting for first notification");
    } else {
      Serial.println("Address: " + xboxController.buildDeviceAddressStr());
      Serial.print(xboxController.xboxNotif.toString());
      unsigned long receivedAt = xboxController.getReceiveNotificationAt();
      uint16_t joystickMax = XboxControllerNotificationParser::maxJoy;
      Serial.print("joyLHori rate: ");
      Serial.println((float)xboxController.xboxNotif.joyLHori / joystickMax);
      Serial.print("joyLVert rate: ");
      Serial.println((float)xboxController.xboxNotif.joyLVert / joystickMax);
      Serial.println("battery " + String(xboxController.battery) + "%");
      Serial.println("received at " + String(receivedAt));
    }
  } else {
    Serial.println("not connected");
    if (xboxController.getCountFailedConnection() > 2) {
      ESP.restart();
    }
  }
  Serial.println("at " + String(millis()));
  delay(500);
}
```

Código alterado com os devidos comandos:

```
#include <XboxSeriesXControllerESP32_asukiaaa.hpp>


//================== MOTORES
int in1 = 27; // aa
int in2 = 26; // ab
int in3 = 25; // ab
int in4 = 33; // bb

// Required to replace with your xbox address
// XboxSeriesXControllerESP32_asukiaaa::Core
// xboxController("44:16:22:5e:b2:d4");

// any xbox controller
XboxSeriesXControllerESP32_asukiaaa::Core xboxController;

void setup() {

    Serial.begin(115200);
  Serial.println("Starting NimBLE Client");
  xboxController.begin();
  pinMode(in1, OUTPUT);
  pinMode(in2, OUTPUT);
  pinMode(in3, OUTPUT);
  pinMode(in4, OUTPUT);
}

void loop() {




  xboxController.onLoop();
  if (xboxController.isConnected()) {
    if (xboxController.isWaitingForFirstNotification()) {
      digitalWrite(2,HIGH);
      Serial.println("waiting for first notification");
    } else {
      int GD = xboxController.xboxNotif.trigRT;
      int GE = xboxController.xboxNotif.trigLT;
      int GCD = xboxController.xboxNotif.btnRB;
      int GCE = xboxController.xboxNotif.btnLB;

      Serial.println("Address: " + xboxController.buildDeviceAddressStr());
      Serial.print(xboxController.xboxNotif.toString());
```

```
 unsigned long receivedAt = xboxController.getReceiveNotificationAt();
 uint16_t joystickMax = XboxControllerNotificationParser::maxJoy;
 Serial.print("joyLHori rate: ");
 Serial.println((float)xboxController.xboxNotif.joyLHori / joystickMax);
 Serial.print("joyLVert rate: ");
 Serial.println((float)xboxController.xboxNotif.joyLVert / joystickMax);
 Serial.println("battery " + String(xboxController.battery) + "%");
 Serial.println("received at " + String(receivedAt));
 // Serial.println(xboxController.xboxNotif.trigRT);
 // Serial.println(xboxController.xboxNotif.trigLT);
 // Serial.println(xboxController.xboxNotif.btnX);
 // Serial.println(xboxController.xboxNotif.btnY);

 float valorGatilho = (float) ((xboxController.xboxNotif.trigRT) * 5 )/1024;
 float PWM = (valorGatilho *255 /5);
 if(GD >0 && GE > 0){


digitalWrite(in1, LOW);
 digitalWrite(in3, LOW);
 digitalWrite(in2, HIGH);
 digitalWrite(in4, HIGH);

 }
  if (GD >500 && GE < 500){
    digitalWrite(in1, LOW);
 digitalWrite(in3, HIGH);
 digitalWrite(in2, HIGH);
 digitalWrite(in4, LOW);
}
 if (GD <500 && GE > 500){
 digitalWrite(in1, HIGH);
 digitalWrite(in3, LOW);
 digitalWrite(in2, LOW);
 digitalWrite(in4, HIGH);
}

 if (GCD == 1 && GCE == 1 ){

  digitalWrite(in1, HIGH);
 digitalWrite(in3, HIGH);
 digitalWrite(in2, LOW);
 digitalWrite(in4, LOW);
}
 if (GD ==0 && GE == 0 && GCD == 0&& GCE == 0){
   digitalWrite(in1, LOW);
 digitalWrite(in3, LOW);
 digitalWrite(in2, LOW);
```

```
      digitalWrite(in4, LOW);
    }
     }
  } else {
    Serial.println("not connected");
          digitalWrite(2,LOW);

    if (xboxController.getCountFailedConnection() > 2) {
      ESP.restart();
    }
  }
  Serial.println("at " + String(millis()));
}
```