

<div><div>Main.cpp</div><div><pre>#include &lt;Arduino.h&gt; #include &lt;SoftwareSerial.h&gt; #include &lt;WiFi.h&gt; #include &lt;TinyGPS++.h&gt; #include &lt;Position.h&gt; #include &lt;Arduino.h&gt;  static const int RXPin = 4, TXPin = 2;  void setup() {    SerialBT.begin("ESP32_Kevin"); //Nom de l'ESP }  void loop() {   position.Position_GPS();   Serial.println("Position GPS: ");   delay(2000);}  </pre></div></div>	<div><div>Position.h</div><div><pre>#ifndef Position_h #define Position_h  #include &lt;Arduino.h&gt;  class Position { private: int pinGPS; Position(int pin); void WIFI; void Position_GPS; };  #endif  </pre></div></div>	<div><div>Position.cpp</div><div><pre>#include "Position.h" #include &lt;SoftwareSerial.h&gt; #include &lt;WiFi.h&gt; #include &lt;TinyGPS++.h&gt;  Position::Position(int pin) { const char* ssid = "Livebox-5520"; const char* password = "53FCFF3DA2A25C6FAFC73FF355"; static const uint32_t GPSBaud = 9600; }  void Position::WIFI() {   Serial.begin(115200);   // Connect to Wi-Fi network with SSID and password   Serial.print("Connecting to ");   Serial.println(ssid);   WiFi.begin(ssid, password);   while (WiFi.status() != WL_CONNECTED) {     delay(500);     Serial.print(".");   }   // Print local IP address and start web server   Serial.println("");   Serial.println("WiFi connected.");   Serial.println("IP address: ");   Serial.println(WiFi.localIP());   {     Serial.begin(115200);     ss.begin(GPSBaud);   } }  string ? à vérifier si c'est bien ce type void Position::Position_GPS() {   {     latitude     while (ss.available() &gt; 0){       gps.encode(ss.read());       if (gps.location.isUpdated()){         Serial.print("Latitude= ");         Serial.print(gps.location.lat(), 6); return(latitude);         Serial.print(" Longitude= ");         Serial.print(gps.location.lng(), 6);         delay(2000);       }     }   } }  </pre></div></div>

il faut modifier la méthode pour qu'elle retourne la latitude, afin de l'afficher après le serial print

il faut décomposer la méthode en 3 méthodes  
1 qui renvoie la latitude = string Position::latitude()  
+1 qui renvoie la longitude = string Position::longitude()  
+ 1 qui renvoie la date = string Position::date()

Ce sont ces valeurs qui seront ensuite à écrire dans BDD

string latitude = ~~Serial.print(gps.location.lat(), 6);~~ return(latitude);