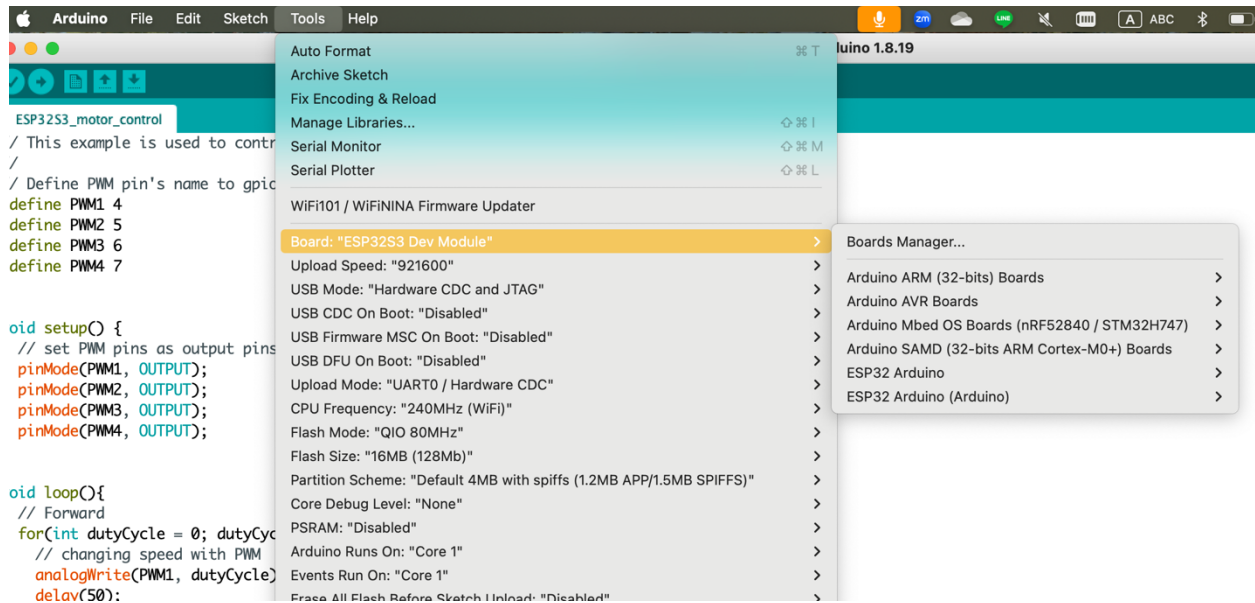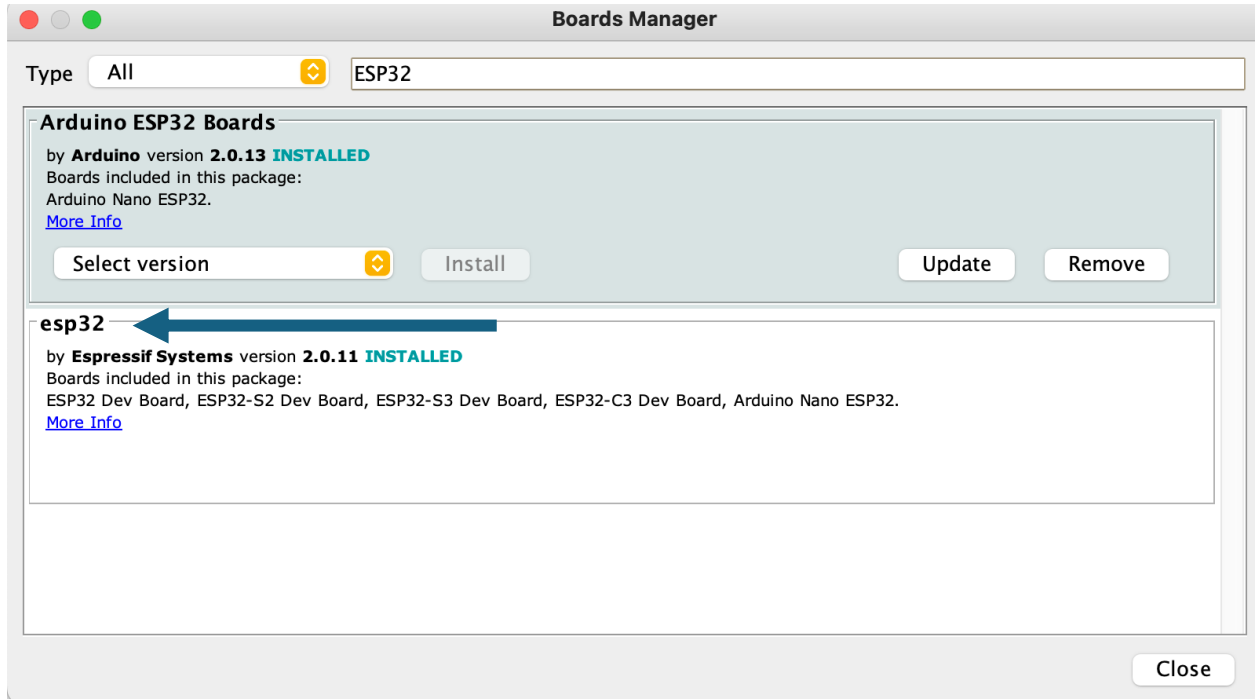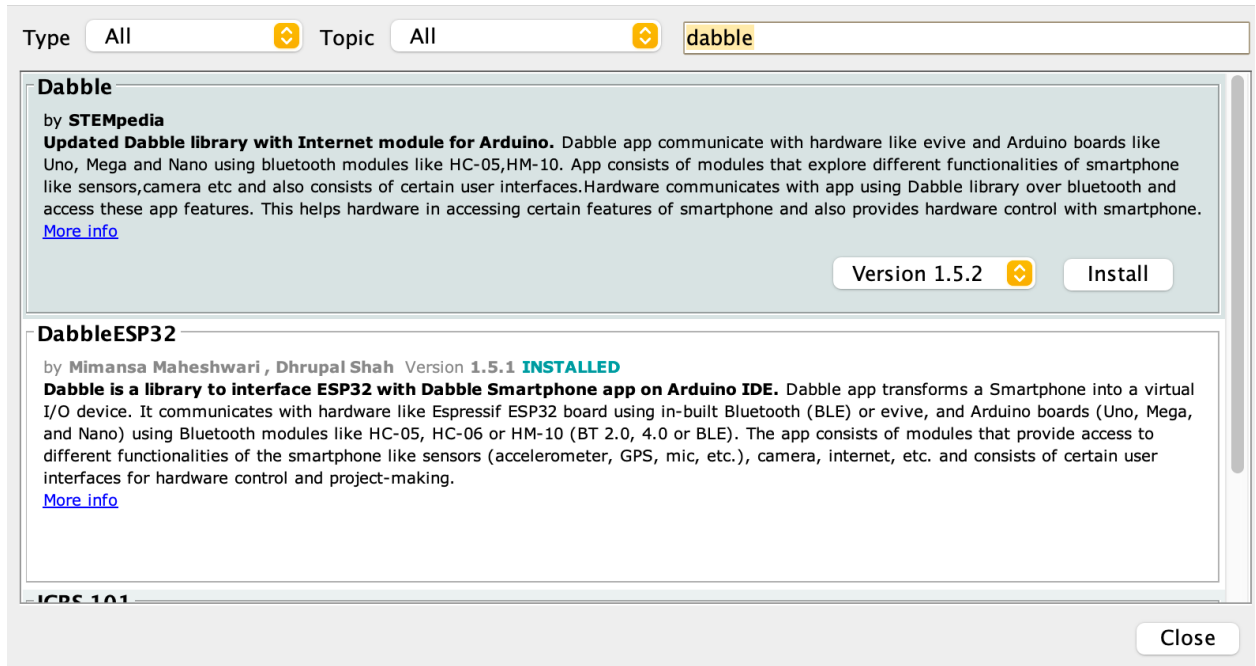To use dabble gamepad
1. Install ESP32 board library. Go to "*Tools*" → "*Board*" →"*Boards Manager...*"

2. Search "ESP32" and install ESP32 board newest version. (

3. Please, go to menu "*Tools*" → "*Manage Libraries*" → search for "*DabbleESP32*"

| Type | All ⬍ | Topic | All ⬍ | dabble |

Dabble

by **STEMpedia**
Updated Dabble library with Internet module for Arduino. Dabble app communicate with hardware like evive and Arduino boards like Uno, Mega and Nano using bluetooth modules like HC-05,HM-10. App consists of modules that explore different functionalities of smartphone like sensors,camera etc and also consists of certain user interfaces.Hardware communicates with app using Dabble library over bluetooth and access these app features. This helps hardware in accessing certain features of smartphone and also provides hardware control with smartphone.
More info

Version 1.5.2 ⬍   Install

DabbleESP32

by **Mimansa Maheshwari , Dhrupal Shah** Version **1.5.1 INSTALLED**
Dabble is a library to interface ESP32 with Dabble Smartphone app on Arduino IDE. Dabble app transforms a Smartphone into a virtual I/O device. It communicates with hardware like Espressif ESP32 board using in-built Bluetooth (BLE) or evive, and Arduino boards (Uno, Mega, and Nano) using Bluetooth modules like HC-05, HC-06 or HM-10 (BT 2.0, 4.0 or BLE). The app consists of modules that provide access to different functionalities of the smartphone like sensors (accelerometer, GPS, mic, etc.), camera, internet, etc. and consists of certain user interfaces for hardware control and project-making.
More info

ICBS 101

Close

4. Try below code or download from

```
#define CUSTOM_SETTINGS
#define INCLUDE_GAMEPAD_MODULE
#include <DabbleESP32.h>
void setup() {
 // put your setup code here, to run once:
 Serial.begin(115200);     // make sure your Serial Monitor is also set at this baud rate.
 Dabble.begin("00MyEsp32-danai");     //set bluetooth name of your device
 pinMode(47,OUTPUT);
 pinMode(48,OUTPUT);
 digitalWrite(47,HIGH);
 digitalWrite(48,HIGH);
 delay(750);
 digitalWrite(47,LOW);
 digitalWrite(48,LOW);
}

void loop() {
 Dabble.processInput();         //this function is used to refresh data obtained from smartphone.Hence calling this function is mandatory in order to get data properly from your mobile.
 Serial.print("KeyPressed: ");
 if (GamePad.isUpPressed())
 {
  Serial.print("Up");
```

```
  digitalWrite(47,HIGH);
  digitalWrite(48,HIGH);
}

if (GamePad.isDownPressed())
{
 Serial.print("Down");
 digitalWrite(47,LOW);
 digitalWrite(48,LOW);
}

if (GamePad.isLeftPressed())
{
 Serial.print("Left");
 digitalWrite(47,HIGH);
 digitalWrite(48,LOW);
}

if (GamePad.isRightPressed())
{
 Serial.print("Right");
 digitalWrite(47,LOW);
 digitalWrite(48,HIGH);
}

if (GamePad.isSquarePressed())
{
 Serial.print("Square");
}

if (GamePad.isCirclePressed())
{
 Serial.print("Circle");
}

if (GamePad.isCrossPressed())
{
 Serial.print("Cross");
}

if (GamePad.isTrianglePressed())
{
 Serial.print("Triangle");
}
```

```
if (GamePad.isStartPressed())
{
  Serial.print("Start");
}

if (GamePad.isSelectPressed())
{
  Serial.print("Select");
}
Serial.print('\t');

int a = GamePad.getAngle();
Serial.print("Angle: ");
Serial.print(a);
Serial.print('\t');
int b = GamePad.getRadius();
Serial.print("Radius: ");
Serial.print(b);
Serial.print('\t');
float c = GamePad.getXaxisData();
Serial.print("x_axis: ");
Serial.print(c);
Serial.print('\t');
float d = GamePad.getYaxisData();
Serial.print("y_axis: ");
Serial.println(d);
Serial.println();
}
```