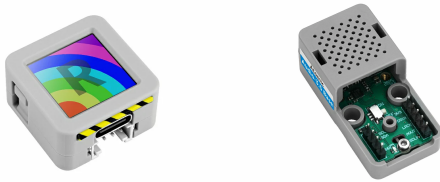


Atomic Audio-3.5 Base Arduino Tutorial

1. Preparation

- 1. Environment setup: Follow the [Arduino IDE Getting Started guide](#) to install the IDE, add board managers for your target board, and install any required driver libraries.
- 2. Required libraries:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Atomic-EchoBase](#)
- 3. Hardware used in this tutorial:
 - [AtomS3R](#)
 - [Atomic Audio-3.5 Base](#)



2. Microphone selection switch

The microphone selection switch on the Atomic Audio-3.5 Base is located as shown below. It selects the microphone input channel to suit different use cases.

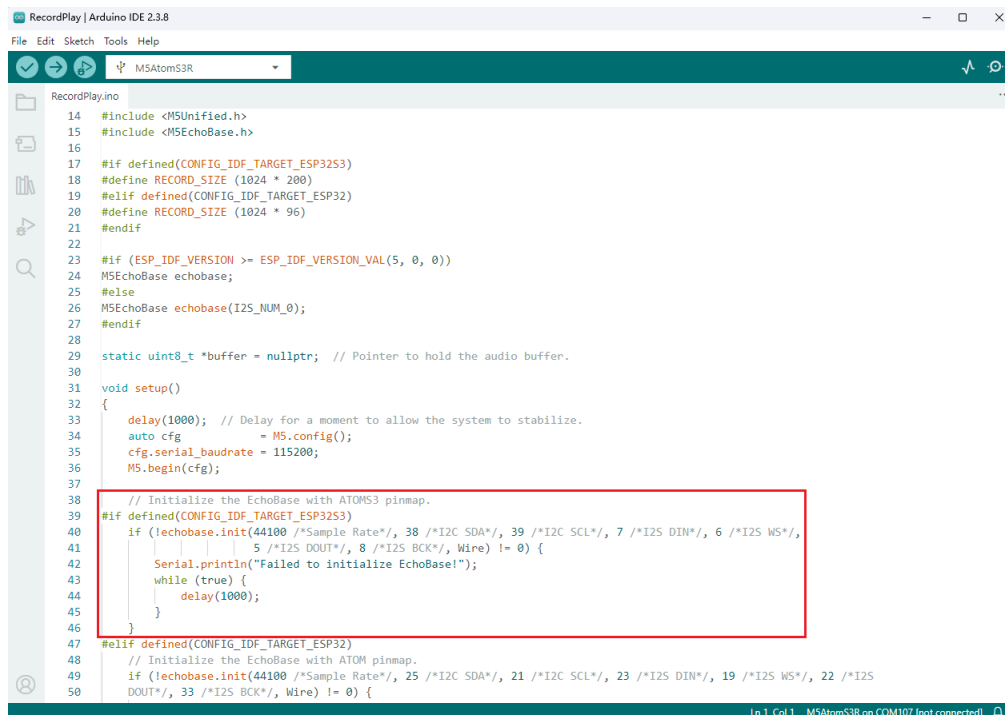
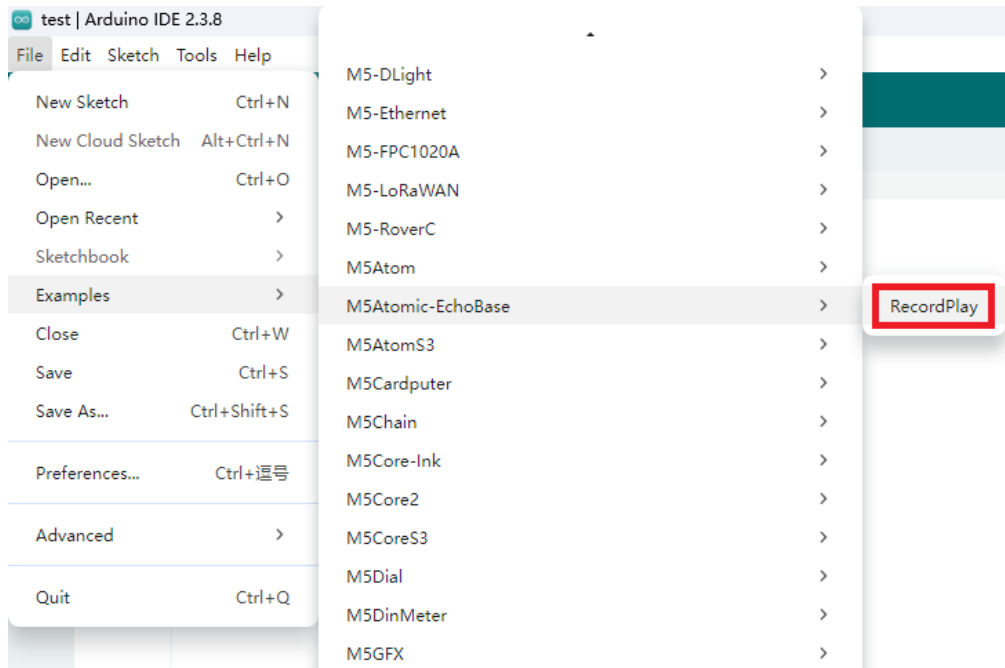


The switch has two positions:

- **ON:** Keep the onboard microphone powered; when external speakers are connected, continue using the built-in microphone.
- **OFF:** When a device is connected to the 3.5mm jack, automatically switch to the external microphone — suitable for headsets or external mic use.

3. Example

Refer to the **RecordPlay** example in the [M5Atomic-EchoBase](#) repository. Adjust IO definitions in the sketch according to your wiring. This tutorial targets AtomS3R; the IO mapping used here matches AtomS3.



Note: The Atomic Audio-3.5 Base microphone supports sample rates between 16 kHz and 64 kHz. Configure a valid sample rate during initialization.

3.1 Initialization

cpp

```
1 // Initialize the EchoBase with ATOMS3 pinmap.
2 echobase.init(44100 /*Sample Rate*/, 38 /*I2C SDA*/, 39 /*I2C SCL*/, 7 /*I2S DIN*/, 6 /*I2S WS*/,
3             5 /*I2S DOUT*/, 8 /*I2S BCK*/, Wire);
4
5 echobase.setSpeakerVolume(50); // Set speaker volume to 50%.
6 echobase.setMicGain(ES8311_MIC_GAIN_6DB); // Set microphone gain to 6dB.
```

3.2 Recording and playback

cpp

```
1 // Recording
2 echobase.setMute(true);
3 delay(10);
4 echobase.record(buffer, RECORD_SIZE); // Record audio into buffer.
5 delay(100);
6
7 // Playing
8 echobase.setMute(false);
9 delay(10);
10 echobase.play(buffer, RECORD_SIZE); // Play audio from buffer.
11 delay(100);
```

3.3 Complete Program

The following example uses `M5Unified` and `M5GFX` to provide a basic display and button control: press the button to record, then automatically play back the recording when finished.

```
1  #include "M5Unified.h"
2  #include "M5EchoBase.h"
3
4  #if defined(CONFIG_IDF_TARGET_ESP32S3)
5  #define RECORD_SIZE (1024 * 200)
6  #elif defined(CONFIG_IDF_TARGET_ESP32)
7  #define RECORD_SIZE (1024 * 96)
8  #endif
9
10 // Create an instance of the M5EchoBase class
11 #if (ESP_IDF_VERSION >= ESP_IDF_VERSION_VAL(5, 0, 0))
12 M5EchoBase echobase;
13 #else
14 M5EchoBase echobase(I2S_NUM_0);
15 #endif
16
17 static uint8_t *buffer = nullptr; // Pointer to hold the audio buffer.
18
19 void setup()
20 {
21     M5.begin();
22     M5.Display.setFont(&fonts::FreeMonoBold9pt7b);
23     Serial.begin(115200);
24
25     #if defined(CONFIG_IDF_TARGET_ESP32S3)
26         // Initialize the EchoBase with ATOMS3 pinmap.
27         if (!echobase.init(44100 /*Sample Rate*/, 38 /*I2C SDA*/, 39 /*I2C SCL*/, 7 /*I2S DIN*/, 6 /*I2S
28             5 /*I2S DOUT*/, 8 /*I2S BCK*/, Wire) != 0) {
29             Serial.println("Failed to initialize EchoBase!");
30             while (true) {
31                 delay(1000);
32             }
33         }
34     #elif defined(CONFIG_IDF_TARGET_ESP32)
35         // Initialize the EchoBase with ATOM pinmap.
36         if (!echobase.init(44100 /*Sample Rate*/, 25 /*I2C SDA*/, 21 /*I2C SCL*/, 23 /*I2S DIN*/, 19 /*I.
37             DOUT*/, 33 /*I2S BCK*/, Wire) != 0) {
38             Serial.println("Failed to initialize EchoBase!");
39             while (true) {
40                 delay(1000);
41             }
42         }
43     #endif
44
45     echobase.setSpeakerVolume(50); // Set speaker volume to 50%.
46     echobase.setMicGain(ES8311_MIC_GAIN_6DB); // Set microphone gain to 6dB.
47
48     buffer = (uint8_t *)malloc(RECORD_SIZE); // Allocate memory for the record buffer.
49     // Check if memory allocation was successful.
50     if (buffer == nullptr) {
```

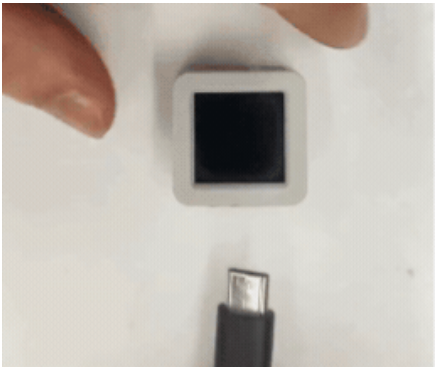
```

51     // If memory allocation fails, enter an infinite loop.
52     while (true) {
53         Serial.println("Failed to allocate memory :(");
54         delay(1000);
55     }
56 }
57
58 Serial.println("Device ready, start recording and playing!");
59 M5.Display.println("Click to \nRecord and Play");
60 }
61
62 void loop()
63 {
64     M5.update();
65     if (M5.BtnA.wasClicked()) {
66         M5.Display.fillScreen(BLACK);
67         M5.Display.setCursor(0, 0);
68         M5.Display.println("Recording");
69         Serial.println("Start recording...");
70         // Recording
71         echobase.setMute(true);
72         delay(10);
73         echobase.record(buffer, RECORD_SIZE); // Record audio into buffer.
74         delay(100);
75
76         M5.Display.println("Playing");
77         Serial.println("Start playing...");
78         // Playing
79         echobase.setMute(false);
80         delay(10);
81         echobase.play(buffer, RECORD_SIZE); // Play audio from buffer.
82         delay(100);
83         M5.Display.println("Done");
84         Serial.println("Done");
85     }
86 }

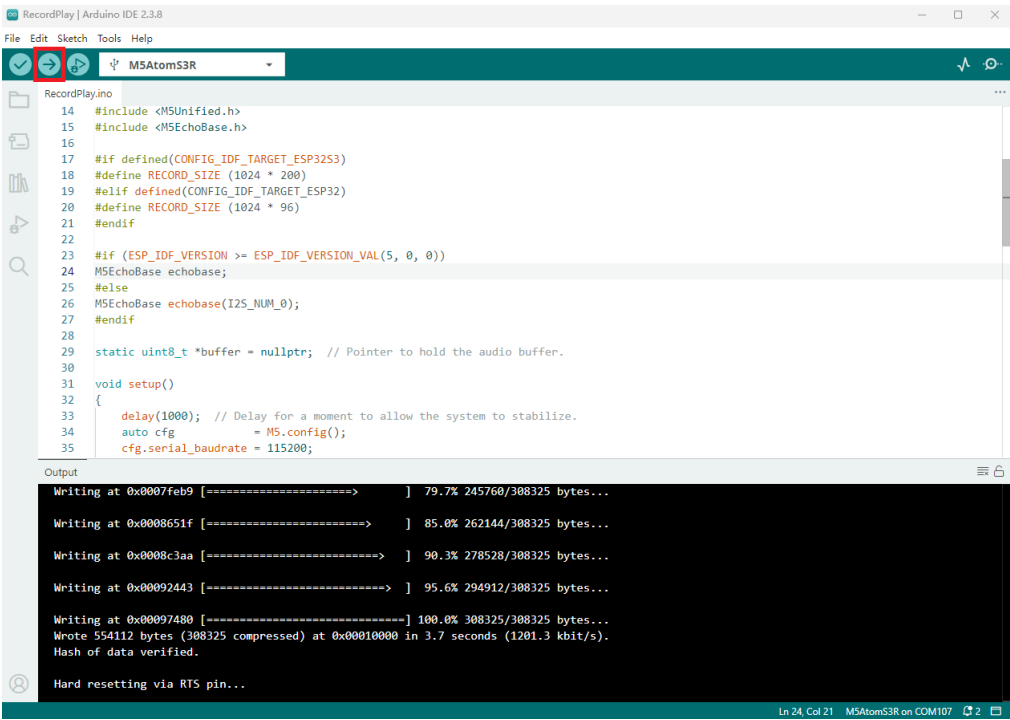
```

4. Compile & Upload

- 1. Download mode: Entering the boot/download mode varies by controller. See the device-specific download instructions at the bottom of the [Arduino IDE Getting Started guide](#).
- For AtomS3R: press and hold the reset button for about 2 seconds until the internal green LED lights, then release — the device enters download mode and is ready for flashing.



- 2. Select the device port and click the Arduino IDE upload button (top-left). Wait for compilation and upload to complete.



5. Recording & Playback

