

Atom VoiceS3R Button

APIs and example programs related to button input on Atom VoiceS3R.

Notes

When using this feature, you need to include the **M5.update()** function in the main loop to read state updates. Also, try to minimize blocking operations, otherwise button changes may not be detected in time.

Example Program

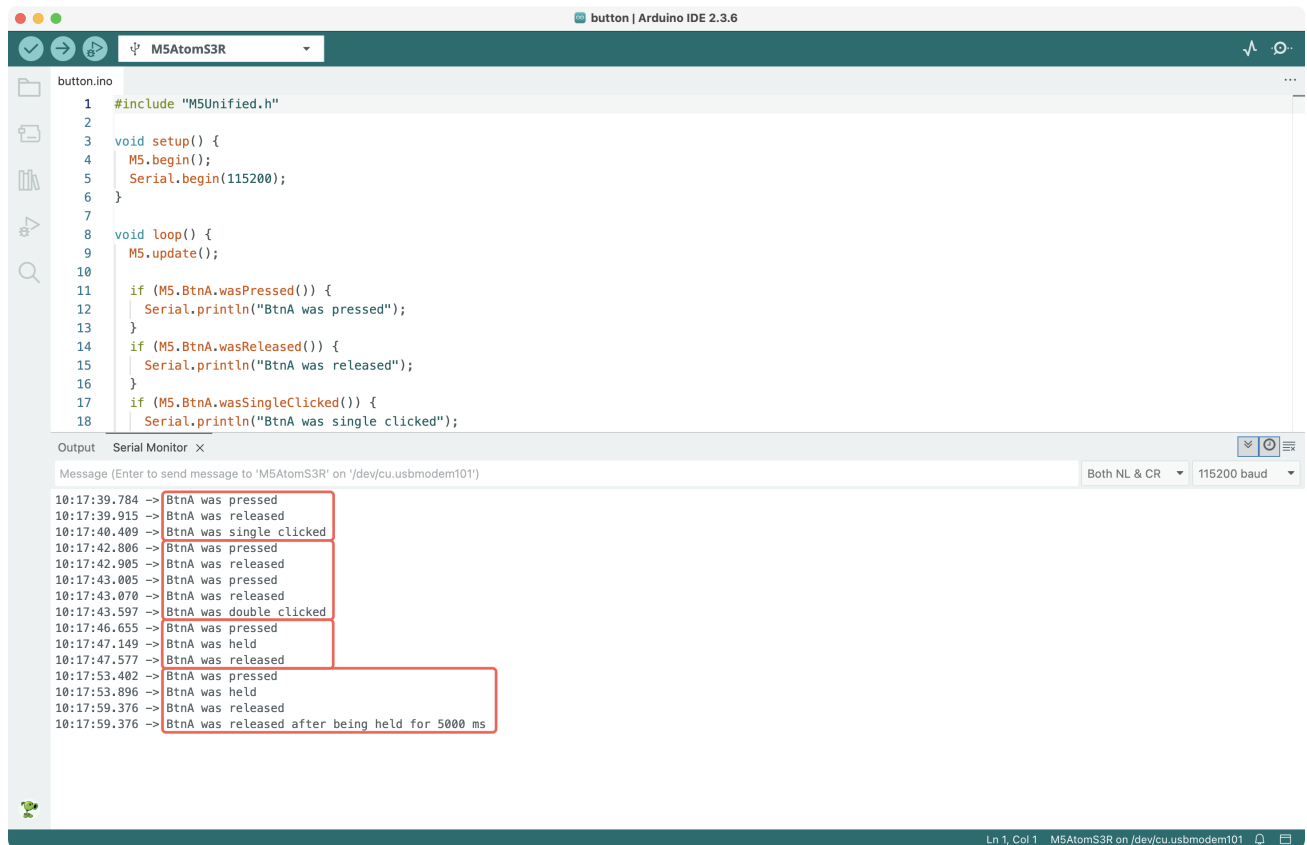
Compilation Requirements

- M5Stack board manager version $\geq 3.2.2$
- Board option = M5AtomS3R
- M5Unified library version $\geq 0.2.8$

cpp

```
1  #include "M5Unified.h"
2
3  void setup() {
4      M5.begin();
5      Serial.begin(115200);
6  }
7
8  void loop() {
9      M5.update();
10
11     if (M5.BtnA.wasPressed()) {
12         Serial.println("BtnA was pressed");
13     }
14     if (M5.BtnA.wasReleased()) {
15         Serial.println("BtnA was released");
16     }
17     if (M5.BtnA.wasSingleClicked()) {
18         Serial.println("BtnA was single clicked");
19     }
20     if (M5.BtnA.wasDoubleClicked()) {
21         Serial.println("BtnA was double clicked");
22     }
23     if (M5.BtnA.wasHold()) {
24         Serial.println("BtnA was held");
25     }
26     if (M5.BtnA.wasReleaseFor(5000)) { // ms
27         Serial.println("BtnA was released after being held for 5000 ms");
28     }
29
30     delay(5);
31 }
```

This program will detect the status of the front button on the device (pressed, released, single short press, double short press, long press, released after a long press of 5000 ms), and print messages in the Serial Monitor:



The screenshot shows the Arduino IDE interface. The top toolbar includes icons for running, uploading, and checking the code. The file explorer on the left shows a folder named 'button.ino'. The main editor displays the code for 'button.ino', which includes the 'M5Unified.h' library and implements the 'setup()' and 'loop()' functions. The 'loop()' function checks for button events: 'wasPressed()', 'wasReleased()', 'wasSingleClicked()', and 'wasDoubleClicked()'. The Serial Monitor at the bottom shows the output of these events, with timestamps and messages like 'BtnA was pressed', 'BtnA was released', 'BtnA was single clicked', 'BtnA was double clicked', 'BtnA was held', and 'BtnA was released after being held for 5000 ms'. The Serial Monitor settings are set to 'Both NL & CR' and '115200 baud'.

```
1 #include "M5Unified.h"
2
3 void setup() {
4   M5.begin();
5   Serial.begin(115200);
6 }
7
8 void loop() {
9   M5.update();
10
11   if (M5.BtnA.wasPressed()) {
12     Serial.println("BtnA was pressed");
13   }
14   if (M5.BtnA.wasReleased()) {
15     Serial.println("BtnA was released");
16   }
17   if (M5.BtnA.wasSingleClicked()) {
18     Serial.println("BtnA was single clicked");
19   }
20   if (M5.BtnA.wasDoubleClicked()) {
21     Serial.println("BtnA was double clicked");
22   }
23   if (M5.BtnA.wasHeld(5000)) {
24     Serial.println("BtnA was released after being held for 5000 ms");
25   }
26 }
```

Serial Monitor Output:

```
10:17:39.784 -> BtnA was pressed
10:17:39.915 -> BtnA was released
10:17:40.409 -> BtnA was single clicked
10:17:42.806 -> BtnA was pressed
10:17:42.905 -> BtnA was released
10:17:43.005 -> BtnA was pressed
10:17:43.070 -> BtnA was released
10:17:43.597 -> BtnA was double clicked
10:17:46.655 -> BtnA was pressed
10:17:47.149 -> BtnA was held
10:17:47.577 -> BtnA was released
10:17:53.402 -> BtnA was pressed
10:17:53.896 -> BtnA was held
10:17:59.376 -> BtnA was released
10:17:59.376 -> BtnA was released after being held for 5000 ms
```

API

The button part of Atom VoiceS3R uses the **Button_Class** from the **M5Unified** library. For more related APIs, please refer to the following documentation:

- [M5Unified - Button Class](#)