

Unit Synth Arduino Tutorial

1. Preparations

- Environment Setup: Refer to the [Arduino IDE Getting Started Guide](#) to complete IDE installation, and install the corresponding board manager and required driver libraries for the development board used.
- Required Driver Libraries:
 - [M5Unified](#)
 - [M5GFX](#)
 - [M5Unit-Synth](#)
- Required Hardware Products:
 - [CoreS3](#)
 - [Unit Synth](#)



2. Notes

Pin Compatibility

Since pin configurations vary across host devices, M5Stack provides a pin compatibility table for reference. Modify the example program according to your actual pin connections.

3. Example Program

- This tutorial uses a CoreS3 as the main controller with a Unit Synth. Unit Synth communicates over UART at a baud rate of `31250`; its corresponding pins after connection are `G18 (RX)` and `G17 (TX)`.

cpp

```

1  #include <M5Unified.h>
2  #include "M5UnitSynth.h"
3
4  M5UnitSynth synth;
5
6  static const uint8_t MELODY_CHANNEL = 0;
7  static const uint8_t DRUM_CHANNEL   = 9;
8
9  void showStatus(const char* stage, const char* detail = nullptr) {
10     // Show the current demonstration stage on the display.
11     M5.Display.fillScreen(TFT_BLACK);
12     M5.Display.setTextColor(TFT_WHITE, TFT_BLACK);
13     M5.Display.setTextSize(2);
14     M5.Display.setCursor(8, 8);
15     M5.Display.println("Unit Synth Test");
16     M5.Display.drawFastHLine(8, 32, M5.Display.width() - 16, TFT_DARKGREY);
17     M5.Display.setCursor(8, 48);
18     M5.Display.setTextColor(TFT_CYAN, TFT_BLACK);
19     M5.Display.println(stage);
20     if (detail != nullptr) {
21         M5.Display.setCursor(8, 82);
22         M5.Display.setTextColor(TFT_WHITE, TFT_BLACK);
23         M5.Display.println(detail);
24     }
25 }
26
27 void playNote(uint8_t channel, uint8_t pitch, uint8_t velocity,
28              uint16_t durationMs) {
29     // Play one note, or wait for a rest.
30     if (pitch == REST) {
31         delay(durationMs);
32         return;
33     }
34
35     synth.setNoteOn(channel, pitch, velocity);
36     delay(durationMs);
37     synth.setNoteOff(channel, pitch, 0);
38 }
39
40 void playScale() {
41     // Play a rising major scale.
42     const uint8_t scale[] = {NOTE_C4, NOTE_D4, NOTE_E4, NOTE_F4,
43                             NOTE_G4, NOTE_A4, NOTE_B4, NOTE_C5};
44
45     for (uint8_t i = 0; i < sizeof(scale); ++i) {
46         playNote(MELODY_CHANNEL, scale[i], 100, 180);
47     }
48 }
49
50 void playChord() {
51     synth.setNoteOn(MELODY_CHANNEL, NOTE_C4, 90);
52     synth.setNoteOn(MELODY_CHANNEL, NOTE_E4, 90);
53     synth.setNoteOn(MELODY_CHANNEL, NOTE_G4, 90);
54     delay(600);
55     synth.setNoteOff(MELODY_CHANNEL, NOTE_C4, 0);

```

```

56     synth.setNoteOff(MELODY_CHANNEL, NOTE_E4, 0);
57     synth.setNoteOff(MELODY_CHANNEL, NOTE_G4, 0);
58 }
59
60 void playDrumPattern() {
61     synth.setInstrument(0, DRUM_CHANNEL, SynthDrum);
62
63     for (uint8_t beat = 0; beat < 8; ++beat) {
64         synth.setNoteOn(DRUM_CHANNEL, 36, 120); // Bass drum
65         synth.setNoteOn(DRUM_CHANNEL, 42, 90); // Closed hi-hat
66         if ((beat & 0x03) == 2) {
67             synth.setNoteOn(DRUM_CHANNEL, 38, 110); // Snare
68         }
69         delay(180);
70     }
71     synth.setAllNotesOff(DRUM_CHANNEL);
72 }
73
74 void configureVoice() {
75     synth.setInstrument(0, MELODY_CHANNEL, ElGrdPiano_3);
76     synth.setVolume(MELODY_CHANNEL, 105);
77     synth.setPan(MELODY_CHANNEL, 64);
78
79     // Program and level are 0..127 values in the MIDI data bytes.
80     synth.setReverb(MELODY_CHANNEL, 1, 55, 20);
81     synth.setChorus(MELODY_CHANNEL, 1, 35, 18, 12);
82
83     synth.setEqualizer(MELODY_CHANNEL,
84                        64, // low band
85                        60, // medium-low band
86                        68, // medium-high band
87                        64, // high band
88                        45, // low frequency
89                        55, // medium-low frequency
90                        70, // medium-high frequency
91                        80); // high frequency
92
93     // 64 is the documented neutral/default tuning value.
94     synth.setTuning(MELODY_CHANNEL, 64, 64);
95     synth.setVibrate(MELODY_CHANNEL, 45, 30, 10);
96     synth.setTvf(MELODY_CHANNEL, 90, 35);
97     synth.setEnvelope(MELODY_CHANNEL, 20, 75, 45);
98
99     synth.setPitchBendRange(MELODY_CHANNEL, 2);
100    synth.setMasterVolume(110);
101 }
102
103 void demonstrateVoiceShape() {
104     // A sustained synth voice makes the filter, envelope and vibrato changes
105     // easier to hear than a short piano note.
106     synth.setInstrument(0, MELODY_CHANNEL, Lead2Sawtooth);
107     synth.setVolume(MELODY_CHANNEL, 110);
108     synth.setPan(MELODY_CHANNEL, 64);
109     synth.setTvf(MELODY_CHANNEL, 48, 100);
110     synth.setEnvelope(MELODY_CHANNEL, 70, 45, 90);

```

```

111     synth.setVibrate(MELODY_CHANNEL, 80, 85, 8);
112
113     playNote(MELODY_CHANNEL, NOTE_C4, 105, 850);
114     playNote(MELODY_CHANNEL, NOTE_G4, 105, 850);
115
116     // Open the filter and shorten the envelope for a second comparison.
117     synth.setTvf(MELODY_CHANNEL, 115, 35);
118     synth.setEnvelope(MELODY_CHANNEL, 10, 85, 35);
119     playNote(MELODY_CHANNEL, NOTE_C5, 105, 850);
120 }
121
122 void demonstrateEffects() {
123     // Use a sustained chord so reverb and chorus tails remain audible.
124     synth.setInstrument(0, MELODY_CHANNEL, StringEnsemble1);
125     synth.setVolume(MELODY_CHANNEL, 115);
126     synth.setReverb(MELODY_CHANNEL, 1, 110, 70);
127     synth.setChorus(MELODY_CHANNEL, 1, 100, 70, 50);
128     synth.setEqualizer(MELODY_CHANNEL, 95, 45, 75, 90, 35, 55, 80, 100);
129
130     synth.setNoteOn(MELODY_CHANNEL, NOTE_C4, 95);
131     synth.setNoteOn(MELODY_CHANNEL, NOTE_E4, 95);
132     synth.setNoteOn(MELODY_CHANNEL, NOTE_G4, 95);
133     delay(900);
134     synth.setNoteOff(MELODY_CHANNEL, NOTE_C4, 0);
135     synth.setNoteOff(MELODY_CHANNEL, NOTE_E4, 0);
136     synth.setNoteOff(MELODY_CHANNEL, NOTE_G4, 0);
137     delay(1200); // Leave time for the effect tails.
138
139     // Tuning is audible when compared with the neutral value.
140     synth.setTuning(MELODY_CHANNEL, 64, 72);
141     playNote(MELODY_CHANNEL, NOTE_A4, 105, 650);
142     synth.setTuning(MELODY_CHANNEL, 64, 64);
143 }
144
145 void demonstratePitchBend() {
146     synth.setNoteOn(MELODY_CHANNEL, NOTE_C4, 110);
147
148     // setPitchBend() maps its input from 0..1023 to the MIDI 14-bit range.
149     synth.setPitchBend(MELODY_CHANNEL, 0);
150     delay(250);
151     synth.setPitchBend(MELODY_CHANNEL, 512);
152     delay(250);
153     synth.setPitchBend(MELODY_CHANNEL, 1023);
154     delay(250);
155     synth.setPitchBend(MELODY_CHANNEL, 512);
156     synth.setNoteOff(MELODY_CHANNEL, NOTE_C4, 0);
157 }
158
159 void demonstrateModWheel() {
160     // Configure the modulation-wheel destinations and depths.
161     synth.setModWheel(MELODY_CHANNEL,
162         20, // pitch
163         30, // TVF cutoff
164         40, // amplitude
165         50, // rate

```

```

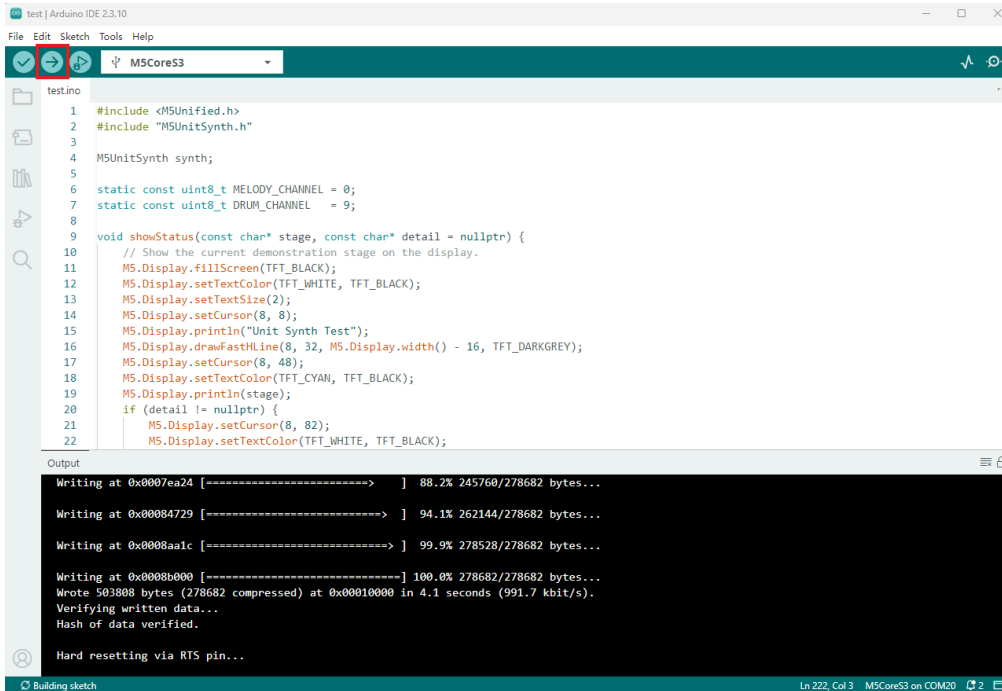
166         25, // pitch depth
167         35, // TVF depth
168         45); // TVA depth
169     }
170
171     void setup() {
172         // Initialize CoreS3 and the MIDI serial port.
173         M5.begin();
174         M5.Display.setTextWrap(true);
175         showStatus("Starting", "MIDI 31250 baud");
176
177         Serial.begin(115200);
178         Serial.println("M5UnitSynth comprehensive example");
179
180         // Change UART pins here if needed.
181         synth.begin(&Serial2, UNIT_SYNTH_BAUD, 18, 17);
182     }
183
184     void loop() {
185         // Repeat the complete feature demonstration.
186         Serial.printf("\n--- M5UnitSynth demo cycle ---\n");
187
188         showStatus("Reset", "Reapply voice settings");
189         synth.reset();
190         delay(100);
191         configureVoice();
192         demonstrateModWheel();
193
194         showStatus("Scale + chord", "Channel 1");
195         Serial.println("Scale and chord");
196         playScale();
197         playChord();
198
199         showStatus("Pitch bend", "C4: low -> high");
200         Serial.println("Pitch bend");
201         demonstratePitchBend();
202
203         showStatus("Voice shaping", "Filter/envelope/vibrato");
204         Serial.println("Filter, envelope and vibrato");
205         demonstrateVoiceShape();
206
207         showStatus("Effects", "Reverb/chorus/EQ");
208         Serial.println("Reverb, chorus, equalizer and tuning");
209         demonstrateEffects();
210
211         showStatus("Drums", "Channel 10");
212         Serial.println("Drums");
213         // Configure drums before sending drum notes.
214         synth.setAllInstrumentDrums();
215         playDrumPattern();
216
217         synth.setAllNotesOff(MELODY_CHANNEL);
218         synth.setAllNotesOff(DRUM_CHANNEL);
219
220         showStatus("Complete", "Next cycle in 1.5 s");

```

```
221     delay(1500);
222 }
```

4. Compile and Upload

- Copy and paste the example code above into the project code area, select the device port (for details, refer to [Compile and Upload](#)), and click the compile and upload button in the upper-left corner of Arduino IDE. Wait for the program to compile and upload to the device.



5. MIDI Synthesizer Feature Demonstration

- After startup, the controller makes Unit Synth continuously play a major scale, a C major chord, pitch bend, filter / envelope / vibrato, reverb / chorus / equalizer, tuning, and drum effects.

