

How to use memory keyer with Bluetooth I/F using Arduino UNO



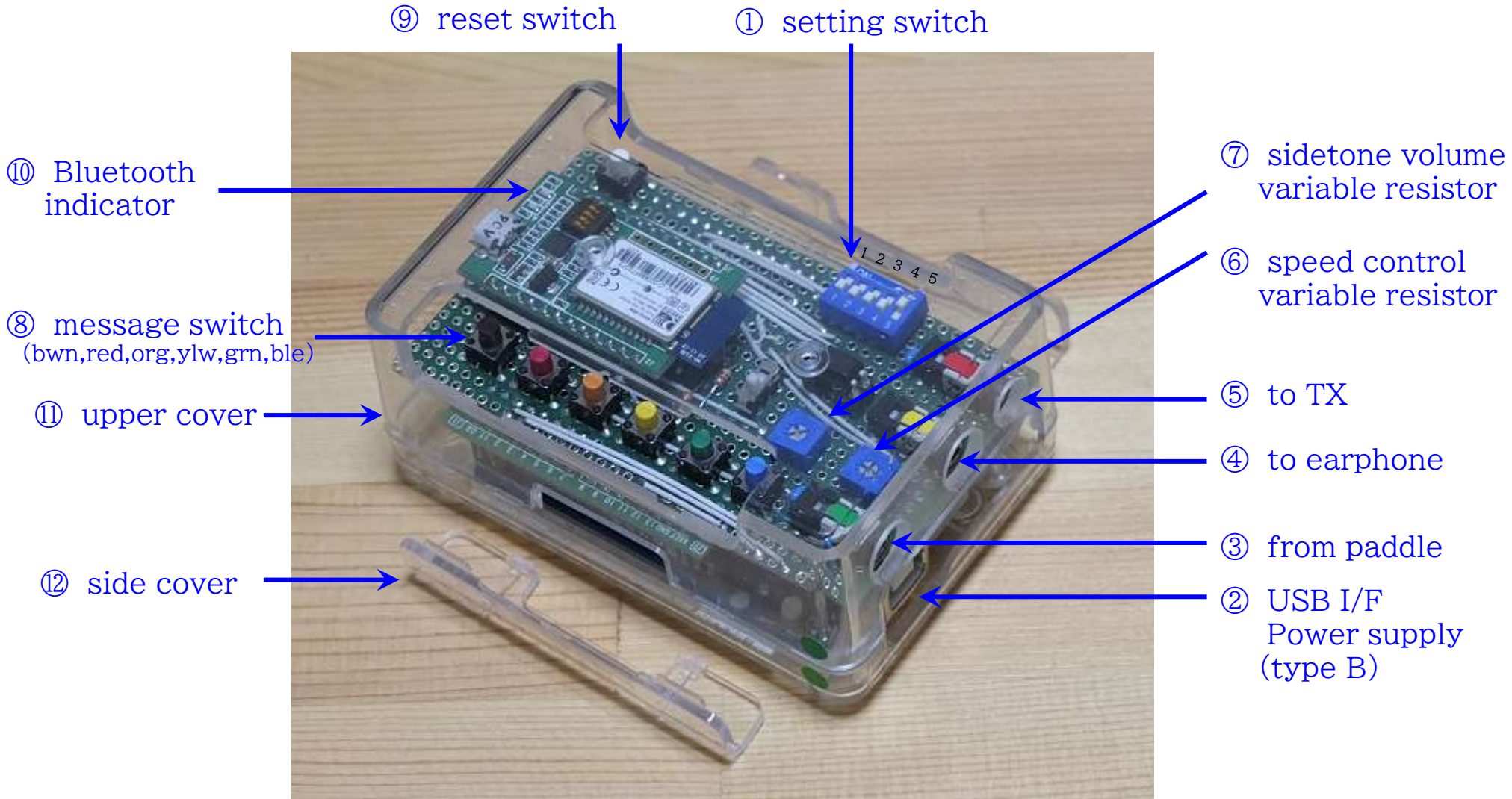
目次

1. Name of each part
2. External connection
3. How to use
 - 3.1 How to use alone
 - 3.2 Control from PC via USB I/F
 - 3.3 Control via Bluetooth I/F from smartphone, tablet terminal, PC etc.
4. How to change the program
 - 4.1 basic matters
 - 4.2 How to modify the program
 - (1) Change sidetone frequency
 - (2) Change speed setting range
 - (3) Change the setting dash, dot, and space
 - (4) Change the setting messages

[Reference]

JA1IWP/JA5IUQ/N2UQ

1. Name of each part



2. External connection (sample.1)



2. External connection (sample.2)



3. How to use

3.1 How to use alone

(1) Setting switch ①(DIP switch 5P)

- **No.1:** set **keyer mode(OFF)** or **Morse practice mode(ON)**

[When using in keyer mode(DIP SW NO.1 OFF)]

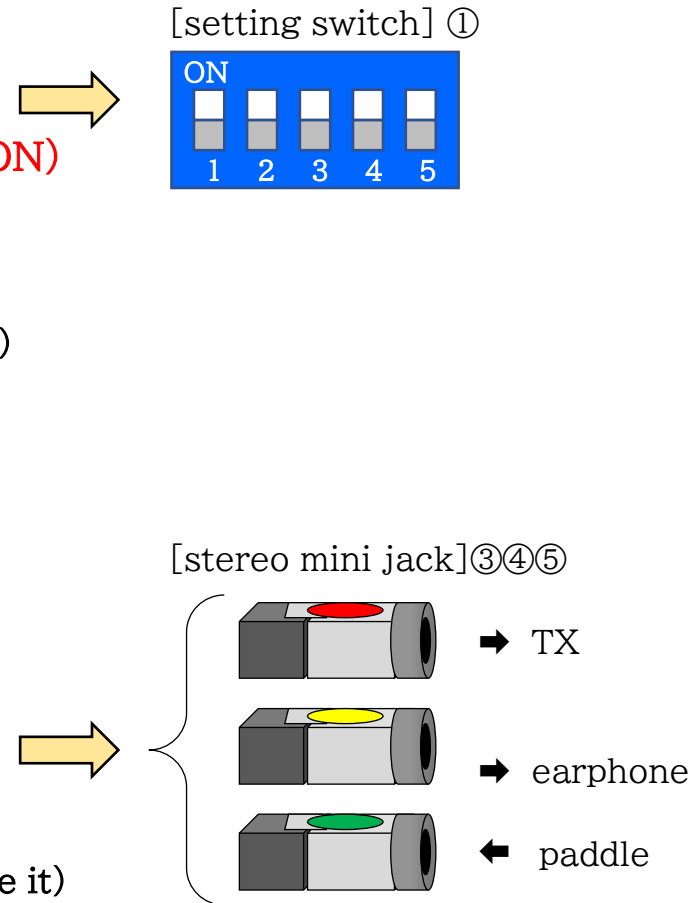
- **No.2:** set memory message set1(OFF), or message set2(ON)
- **No.3:** set electric keyer mode(OFF), or straight key mode(ON)
- **No.4:** set paddle-normal(OFF), or paddle-reverse(ON)

[When using in Morse practice mode(DIP SW NO.1 ON)]

- **No.2:** set European mode(OFF), or Japanese mode(ON)
- **No.3, No.4:** set random patterns(1-4)
- **No.5:** set sidetone off(OFF), or sidetone on(ON)

(2) External connection

- **power supply:** input to Arduino Uno USB-B type connector②
- **paddle:** connect to 3.5mm stereo mini jack ③ (green)
- **earphone:** connect to 3.5mm stereo mini jack ④ (yellow)(when use it)
- **TX:** connect to 3.5mm stereo mini jack ⑤ (red)

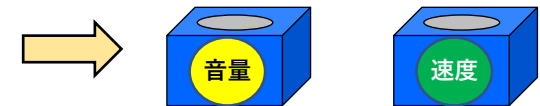


3. How to use

3.1 How to use alone (continued)

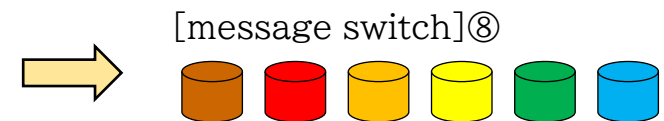
(3) Speed control, Sidetone volume control

- **speed control variable resistor**⑥ (right-side, green):
Turn clockwise to speed up
- **sidetone volume variable resistor**⑦ (left-side, yellow):
Turn clockwise to volume up



(4) Set memory messages

- Press message switch⑧ (tact SW) (brown(ch.1)~cyan(ch.6))
- message11~16 will be generated (when setting SWNo.2 OFF)
- message21~26 will be generated (when setting SWNo.2 ON)
- Press reset switch to abort⑨ (tact SW:white)



(5) Reset on all modes

- Press reset switch to abort⑨ (tact SW:white)



3. How to use

3.2 Control from PC via USB I/F

[using Arduino IDE serial monitor function]

- (1) Start Arduino IDE
- (2) [tool]→ [board]→ [Arduino AVR boards]→ [Arduino Uno]
- (3) [tool]→ [serial port]→ (Arduino Uno)
- (4) [tool]→ [serial monitor] open [serial monitor]
- (5) set [speed] to 「115200bps」
- (6) In input column, set characters to generate, then [send]
 - To generate message11~16, send [Z],[X],[C],[V],[B],[N] respectively
 - To generate message21~26, send [A],[S],[D],[F],[G],[H] respectively
 - Press reset switch to abortⓄ (tact SW:white)
- (7) speed control
 - [+],[++],[+++] - - - input make speed increase
 - [-],[--],[---] - - - input make speed decrease

3. How to use

3.2 Control from PC via USB I/F

[Using Tera Term]

- (1) Start Tera Term
- (2) On the "Tera Term: New Connection" , select "Serial (E)", and for "Port (R)", select "COM ◎" where (Arduino Uno) is visible and click "OK".
- (3) Select "Settings" → "Serial Port" to display the "Serial Port Settings and Connections" screen. Set "Speed" to "115200" and "Reset current connection".
- (4) When you enter the characters you want to generate (lowercase letters, numbers, symbols) from the keyboard, Morse code is generated in sequence.
 - To generate messages 11 to 16, enter "Z", "X", "C", "V", "B", and "N" at the bottom of the keyboard.
 - To generate messages 21 to 26, enter "A", "S", "D", "F", "G", and "H" in the middle of the keyboard.
 - To cancel, press the reset switch (tact SW: white) on the main unit.
- (5) speed control
 - [+],[++],[+++] - - - input make speed increase
 - [-],[--],[---] - - - input make speed decrease

3. How to use

3.3 Control via Bluetooth I/F from smartphone, tablet terminal, PC, etc.


[tablet terminal(Fire HD8) and 「Bluetooth Terminal」(free App)]

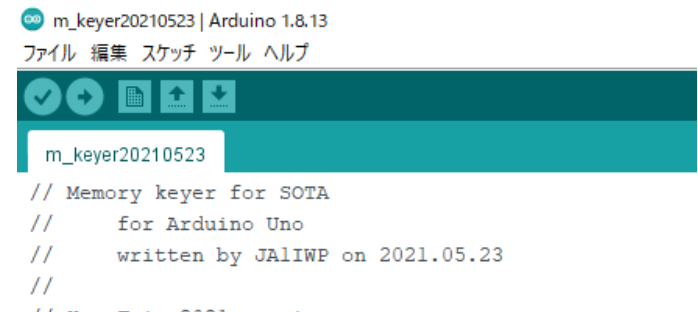
- (1) タブレット端末の「設定」で、「接続デバイス」から「Bluetooth」を「ON」にしておく
- (2) Bluetoothの設定で、キーヤー本体搭載のBluetoothモジュール「RNBT-6BA4」とペアリングしておく
- (3) 「Bluetooth Terminal」を起動
- (4) 右上の「Menu」→「Select Device」から「RNBT-6BA4」を選択
「RNBT-2D73」のBluetoothインジケータ⑩が、点滅から点灯になったことを確認する
- (5) 入力欄に、発生させたい文字(アルファベット小文字、数字、記号)を入力し「Send」
 - ・ メッセージ11～16の発生は、キーボード下段の「Z」「X」「C」「V」「B」「N」を入力し「Send」
 - ・ メッセージ21～26の発生は、キーボード中段の「A」「S」「D」「F」「G」「H」を入力し「Send」
 - ・ 中止する場合は、本体のリセット・スイッチ(タクトSW:白色)を押下
- (7) 速度調節
 - ・ 「+」「++」「+++」..を入力すれば、速度増加 (「++Z」等の入力も可能)
 - ・ 「-」「--」「---」..を入力すれば、速度低下
- (8) 終了時は、「Menu」→「Disconnect」とし、「RNBT-2D73」のインジケータが点滅に戻ったことを確認す
- (9) 「Bluetooth Terminal」は、これ自体にメッセージを設定できる機能がある(4ch分)
 - ・ 「Menu」→「Settings」とすれば、メッセージ設定画面が現れるので、各々設定し「Save」して使用

4. How to change the program(sketch)

4.1 basic matters

・ Connect Arduino Uno to a PC with Arduino IDE installed with a USB cable

- (1) Arduino IDE を起動し、「ファイル」→「開く」あるいは「最近使った項目を開く」から、変更したいファイル(スケッチ)を開く
- (2) 「ファイル」→「名前を付けて保存」で、違う名前で保存する
- (3) 「ツール」→「ボード」→「Arduino AVR boards」→「Arduino Uno」を選択
- (4) 「ツール」→「シリアルポート」は(Arduino Uno)が見えているCOM【番号】を選択する
- (5) 下記の方法で変更後、「スケッチ」→「検証・コンパイル」でコンパイルを実施
 - ・あるいは、左上の マークをクリック
 - ・エラーメッセージ(赤字)が出たら、修正箇所を確認する(多分、入力ミス(全角文字を使ったとか、行の最後に「;」を付けなかったとか))
- (6) コンパイルに問題なければ、「スケッチ」→「マイコンボードに書き込む」で実行ファイルをキーヤ本体にアップロードする(あるいは、左上の  マークをクリック)
- (7) 動作を確認し、OKならば「保存」して終了



The screenshot shows the Arduino IDE interface. At the top, the title bar reads "m_keyer20210523 | Arduino 1.8.13". Below the title bar, there are menu items: "ファイル", "編集", "スケッチ", "ツール", "ヘルプ". A toolbar contains several icons, including a green checkmark icon. Below the toolbar, a code editor window is open, displaying the following code:

```
m_keyer20210523
// Memory keyer for SOTA
//   for Arduino Uno
//   written by JAlIWP on 2021.05.23
//
```

4. How to change the program(sketch)

4.2 How to modify the program

(1) Change the sidetone frequency

- Search by " **int Sidetone_freq** " from "Edit" → "Search"

```
int Sidetone_freq = 587;
```

Change the variable that defines the frequency (set to 587Hz in the example) to the frequency (Hz) you like, and compile and upload as Section 4.1.

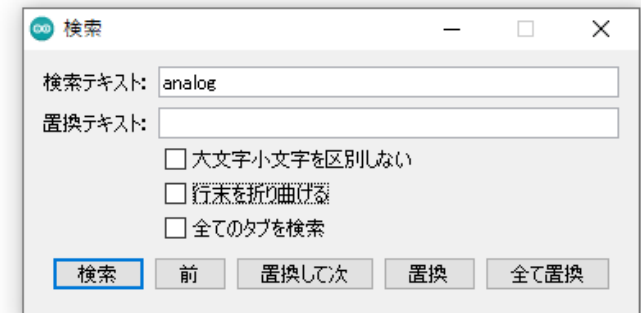
(2) Change speed setting range

- Search by "**analog**" from "Edit" → "Search"

- **val = analogRead(A0);**

```
T = (1024 - val) / 32 + 8 + offset;
```

- The variable val is the AD-converted value (0 to 1023) of the voltage of the speed adjustment volume.
- The variable T is the basis of the delay time. Here, "32" determines the slope (speed change amount) and "8" determines the shortest delay time, so change it as appropriate.
- In this example, "val" changes from 0 to 1023, while "T" changes from 40ms to 8ms.
- Variable offset is a variable for speed control with "+" or "-" from the outside.



4.2 How to modify the program (continued)

(3) Change the setting of dash/dot/character spacing

- Search by "analog" from "Edit" → "Search"

- `val = analogRead(A0);`

- `T = (1024 - val) / 32 + 8 + offset;`

- `T3 = T * 6;`

- `T4 = T * 8;`

- `T = T * 2;`

- The length of the long point is specified by "T3", the character spacing is specified by "T4", and the length of the short point is specified by "T". For example, if you want to set the long point to 3.5, $T3 = T * 7$; Change to, change to $T4 = T * 9$; if you want the character spacing to be 4.5
- If you want to change it more precisely, change "32" from the above "T" setting formula by making it smaller.

4.2 How to modify the program (continued)

(4) change the setting messages (continued)

- For example, change message 11 (hereinafter, the same procedure up to message 26)
- Search by "MSG11" from the "Edit"-> "Search" screen.

```
void MSG11() {  
  CQ(); CQ(); CQ(); DE();  
  jaliwpsla3(); JCC(); N2703();  
  m_k(); delay(T4);  
}
```

Go to and change the inside of {} as appropriate. For example, if you want to set it to "8J1HAM"

```
void MSG11() {  
  m_8(); m_j(); m_l(); m_h(); m_a(); m_m(); delay(T4);  
}
```

- However, since it will be long as it is, frequently used parts are defined separately. For example, if it is "CQ" (search by "CQ"!)

```
void CQ() {  
  m_c(); m_q(); delay(T4);  
}
```


4.2 How to modify the program (continued)

(4) change the setting messages (continued)

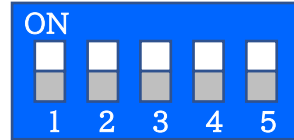
Furthermore, [JA1IWP/3]

```
void jaliwp() {  
  m_j(); m_a(); m_l(); m_i(); m_w(); m_p(); delay(T4);  
}
```

```
void jaliwpsla3() {  
  jaliwp(); m_sla(); m_3(); delay(T4);  
}
```

[Reference]

[setting switch]①



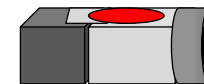
1	Setting switch 1	OFF	Keyer mode
2	Setting switch 2	OFF	Message set 11~16
3		ON	Message set 21~26
4	Setting switch 3	OFF	エレクトロニクスキー・モード
5	Setting switch 4	OFF	Paddle normal
6		ON	Paddle reverse
7	Setting switch 3	ON	Strait key mode
8	Setting switch 1	ON	Morse practice mode
9	Setting switch 2	OFF	European mode
	Setting switch 3	-	Set random patterns (1~4)
	Setting switch 4	-	
10	Setting switch 2	ON	Japanese mode
11	Setting switch 3	-	Set random patterns (1~4)
12	Setting switch 4	-	
13	Setting switch 5	OFF	Sidetone OFF
		ON	Sidetone ON

[message switch]⑧

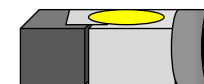


	Setting switch 2 OFF	Setting switch 2 ON
1	message 11	message 21
2	message 12	message 22
3	message 13	message 23
4	message 14	message 24
5	message 15	message 25
6	message 16	message 26

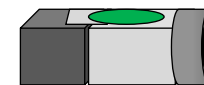
[stereo mini jack]



➔ to TX ⑤



➔ to earphone ④



➔ Form paddle ③