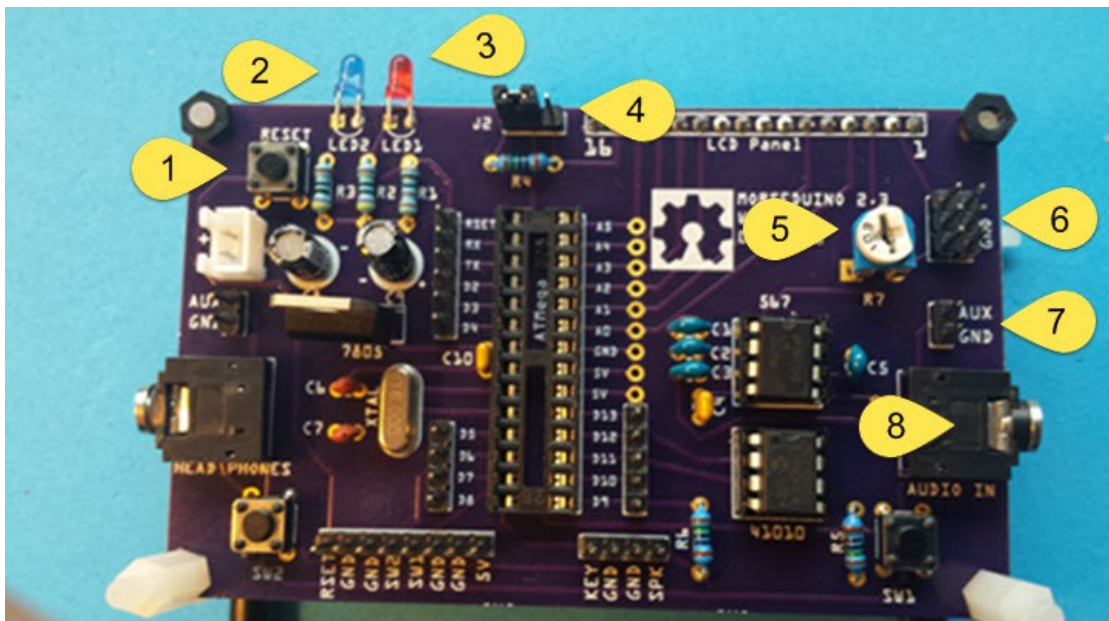
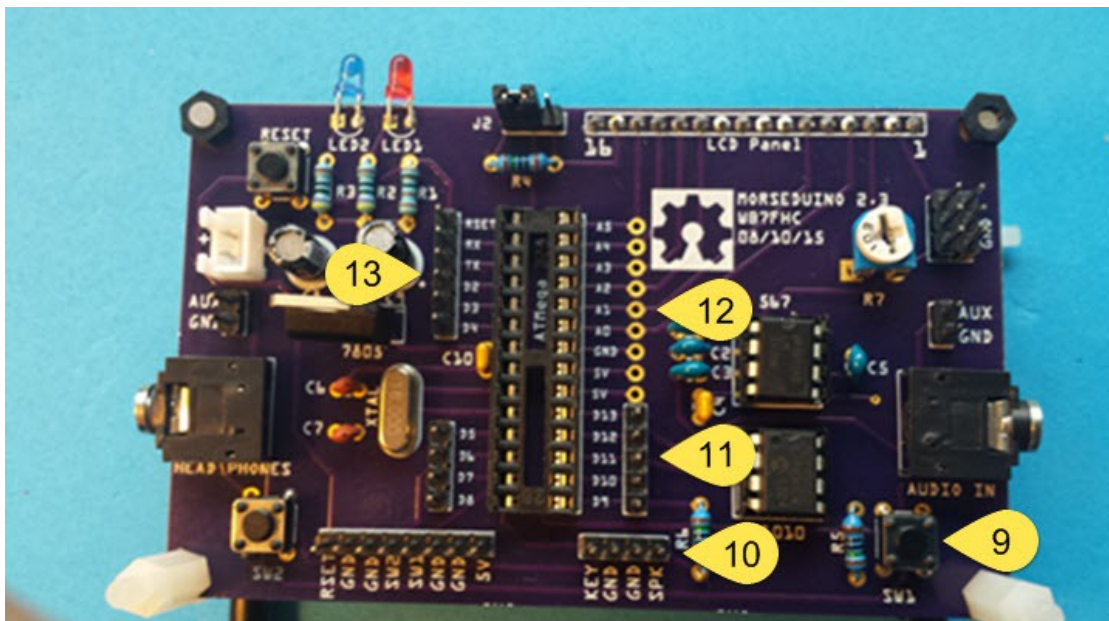


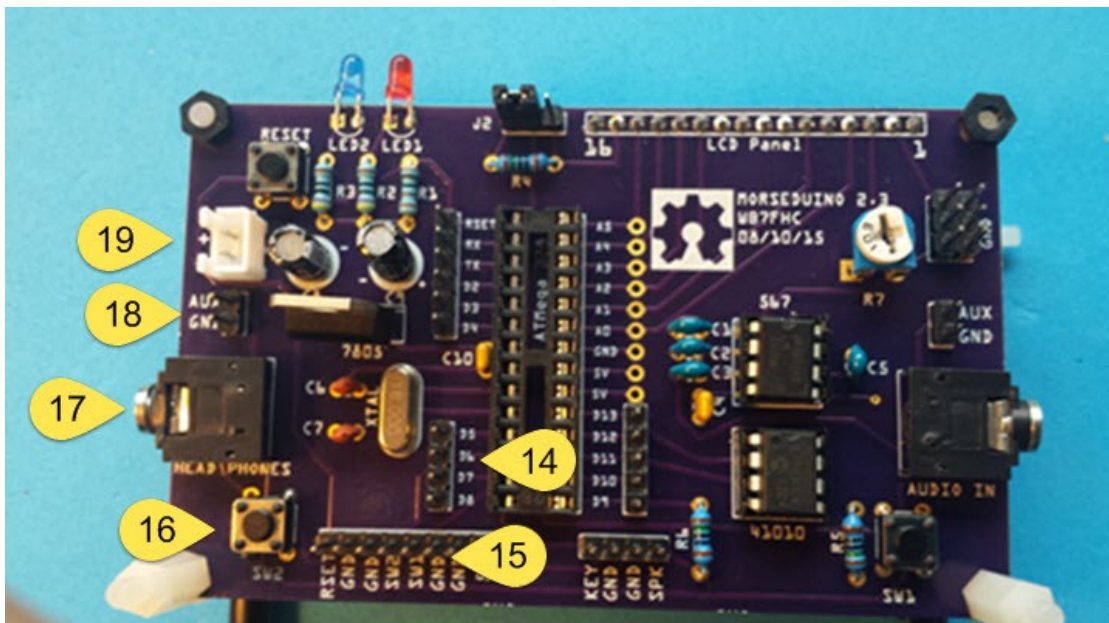
## M2 Walk Around



1. RESET ... This button resets the ATmega 328. If the decoder does not seem to print the correct letters, press this button. If you connect your speaker while the unit is powered, press this button.
2. LED2 ... Blue. This LED is driven by D5 on the ATmega. The program will turn this on for the duration of what it thinks is a valid dit or dah. The LED lights when D5 is LOW.
3. LED1 ... Red. This LED is driven by D8, which is pulled down by pin 8 on the LM567 tone decoder. This LED will flicker when there is noise on the band. It will also stay on when LED2 is on. So it, too, blinks with the CW. Noise can be observed when it flickers and the blue LED does not light.
4. J2 ... This jumper provides for two levels of brightness on the LCD back light. When the left two pins are jumped 5V goes through R4 (150 ohms) to pin 15 of the LCD. This dims the back light. When the jumper is to the right, 5V goes directly to pin 15 and the back light is bright.
5. R7 ... This is the contrast adjustment for the LCD panel.
6. 5V/GND ... These header pins provide an easy way for you to tap Vcc and GND from your board. You can use them to power other devices that draw minimal current. If you have an alternate source of power you can use these pins to provide Vcc instead of using the board's own voltage regulator circuit. If you want to program your ATmega chip while it is on the board with an Arduino Uno, disconnect power at Vin and connect these pins to 5V and GND on the Arduino. (see notes on RSET, RX and TX #13 below).
7. AUX/GND ... These pins, located on both sides of the board, provide you an alternate means of connecting audio to the board. If you want to place the M2 in an enclosure, these pins can be used to connect external jacks.
8. AUDIO IN ... The two head phone jacks on opposite sides of the board are wired together. Either of these jacks can be used to bring audio from your radio into the M2.



9. SW1 ... This button which will be on the left hand side of the unit when you are facing the LCD serves two functions depending on whether or not a speaker is attached. **When no speaker is connected**, This button can be used to tell the ATmega to change the resistance on the digital potentiometer 41010. This will change the frequency of the audio tone the decoder will seek. A value will be displayed on the LCD as you press this button. The value is not the frequency but an indication for where in the range between 32 Hz and 3200 Hz the sweet spot sits. The higher the number the higher the frequency. To switch between increasing and decreasing the value, simply pause for 2 seconds between button presses. **When a speaker and key are connected**, sending the Morse Code sequence: dah-dit-dit-dah-dah will let you use this button to decrease the frequency of the CW practice side tone. The actual frequency of the tone will be displayed on the LCD. A range of 110 Hz to 990 Hz is possible in steps of 110. SW1 and SW2 can be used to change the frequency of the tone. Send the same sequence to exit this routine and the button will revert to normal. The tone you select is stored and will be recalled on reset or power up.
10. KEY/GND/GND/SPK ... Use these pins to hook a speaker and key to the unit. If you plan to put the M2 in an enclosure these will make it easy to run wires to jacks.
11. D9 – D12 ... These are ATmega I/O pins. They are labeled using normal Arduino designation. D9 connects to the speaker pin through R6 (150 ohms). The following pins manage the SPI bus used to communicate with the digital potentiometer (41010). D10:SS, D11:MOSI, D12:MISO, D13:SCK. These pins could be used for ICSP but the 41010 should be removed during programming.
12. A0 – A5 ... The analog pins are being used as digital output pins the drive the LCD panel. It may be unfortunate that the evolution of this design has grabbed all of the analog pins for this purpose leaving no analog pins for custom applications. If person wanted to use one of the analog pins a trace could be cut and a jumper run from one of the unused I/O pins.
13. RSET – D8 ... These pins correspond to Arduino designation. RX and TX can be used as D0 and D1 in custom applications. If you want to program the ATmega while it is in the board, remove the micro-controller chip from an Arduino Uno and the power plug from the M2. Link 5V and GND between the two boards and run jumpers between RSET, RX and TX on both boards. Attach the USB plug to the Uno and program the chip on the M2. All will behave as if your chip was installed on the Arduino. D2, D3, and D4 are not used on the M2 and are available for your own custom purposes.



14. D5 – D8 ... These pins correspond to Arduino designation. D5 is OUTPUT and drives LED2 and D8 is the INPUT from the LM567 tone decoder and also drives LED1. You can hook oscilloscope leads to these pins to observe the software managed noise filter. Hooking the oscilloscope to the cathodes of the two LEDs will do the same. D6 and D7 are connected to SW2 and SW1 respectively.
15. RSET – 5V ... All of these pins are here to provide an easy way to hook external switches should a person want to install the M2 in an enclosure. Note that this includes yet another way to bring in (or tap out) regulated 5V and GND.
16. SW2 ... This button which will be on the right hand side of the unit when you are facing the LCD serves multiple functions depending on whether or not a speaker is attached. **When no speaker is connected, A quick press** of this button will increment the software noise filter. The value you see on the LCD indicates a delay in milliseconds that the program will wait after detecting a tone when D8 is pulled low, before it tests a second time to see if the pin is still low. This allows the decoder to ignore noise spikes that otherwise would result in misread CW. These can be viewed graphically on an oscilloscope connected as described above. Possible settings are from 0 – 8 milliseconds where 0 is no filtering. Pressing the button only moves the value up, but it rolls back to 0 after 8. **Holding the button down longer** will initiate an automatic sweep of the audio frequency sweet spot in the tone decoder. The words 'Sweep Up' and 'Sweep Down' will appear on the LCD along with dots when a tone is detected. The M2 will determine the highest and lowest settings that decode a tone and then park itself halfway between the two. Each pass up and down will go slower than the one before it. You can abort the sweep routine by pressing the RESET button. **When a speaker and key are connected,** sending the Morse Code sequence: dah-dit-dit-dah-dah will let you use this button to increase the frequency of the CW practice side tone. The actual frequency of the tone will be displayed on the LCD. The range of 110 Hz to 990 Hz is possible in steps of 110. The tone you select is stored and will be recalled on reset or power up.
17. HEAD PHONES ... is identical to #8, AUDIO IN. You can hook head phones or ear buds to either of these jacks to hear the CW from your receiver. These jacks and both AUX/GND headers are all interchangeable.
18. AUX/GND ... See 17 above.
19. Vin ... Voltage In. 7 to 14.5 VDC. Note the circuit is not reverse voltage protected. The positive pin is marked with a '+'.