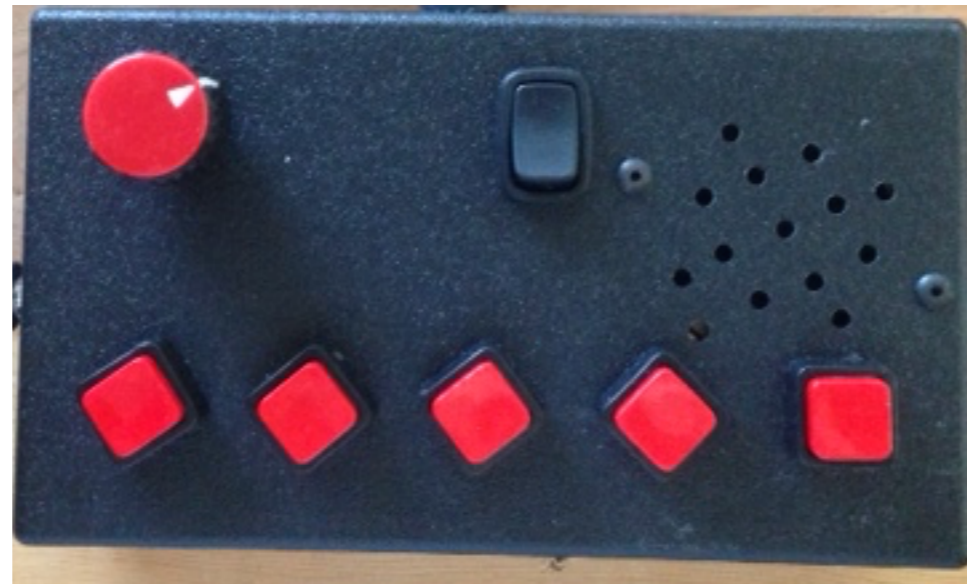


lambic Memory Keyer

Clay Davis, AB9A
Designer, Programmer, Builder

Ray Day, N6HE
Builder, Beta Tester and Feature Imagineer



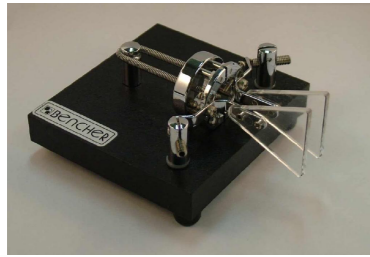
Morse Code vs CW

- Morse Code = A method of encoding letters, numbers etc, using dits and dahs.
- CW = Continuous Wave, a modulation type.

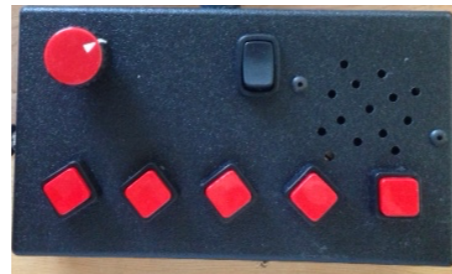
Morse Code is alive & well

- If you want to reach out and touch the world, you want to be on HF.
- As many CW QSOs as SSB QSOs
- Sometimes DX only available on CW
- Your ear can pick out a CW “tone” at signal to noise levels that make Phone contacts unintelligible.

Paddle

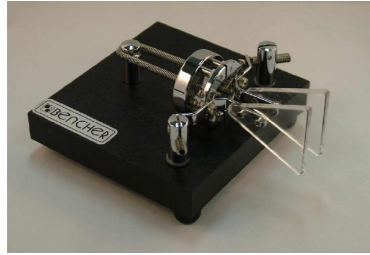


Keyer

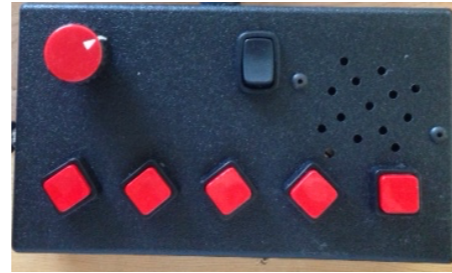


- Accepts contact closures from paddle and send precisely timed code to radio.
- Reduces the number of Paddle movements required to send Morse code.
- Plays prerecorded messages with a single key press.
- Journey from straight key to lambic Memory Keyer is about a progressively more efficient user interface and less operator fatigue.

Paddle



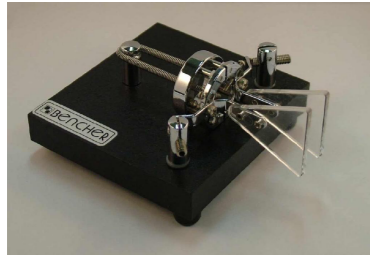
Keyer



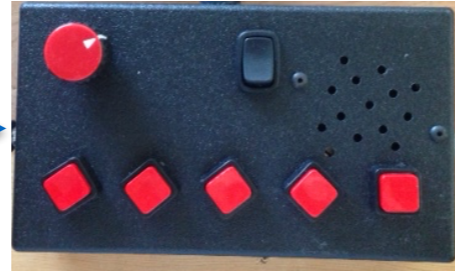
Goals:

- No Menus, all front panel controls
- Program messages with a Paddle
- Program messages without a Radio
- Clean side tone with speaker

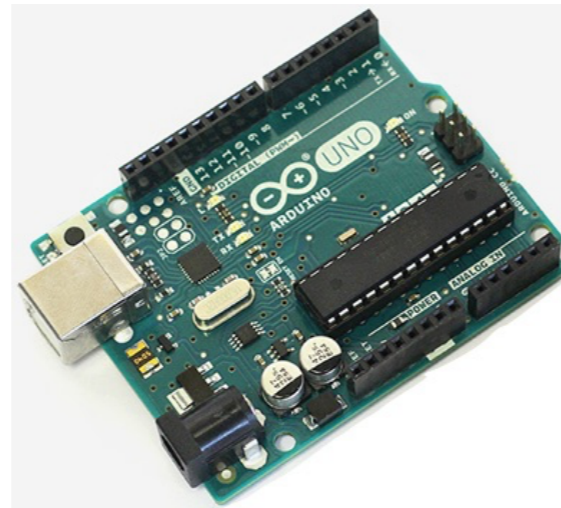
Paddle



Keyer



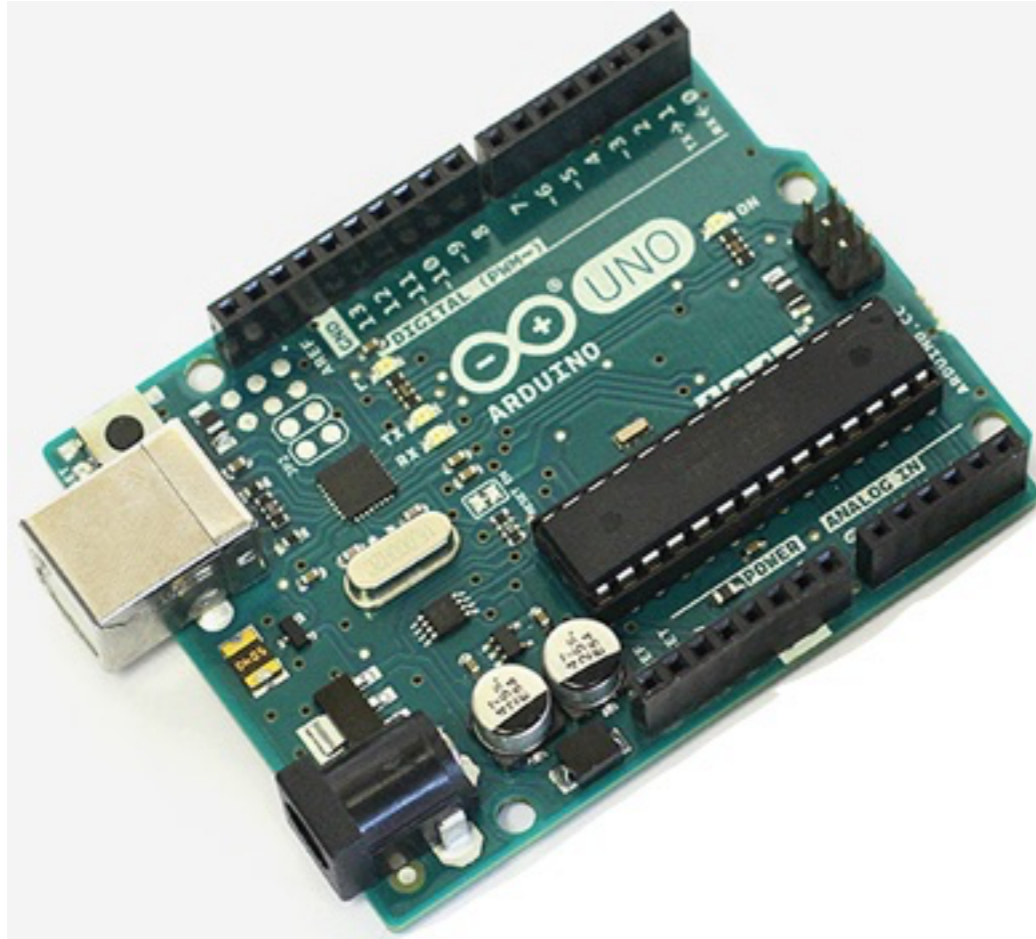
+



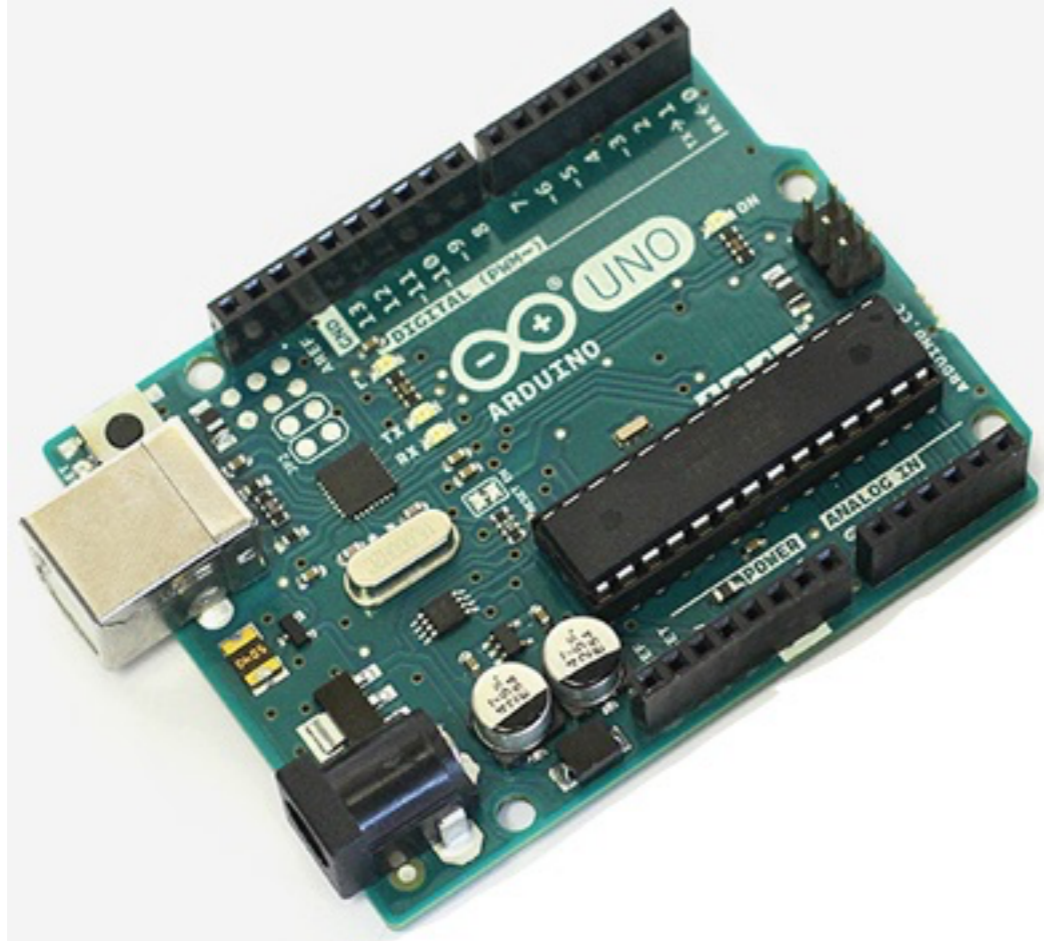
+

```
sketch_AB9A_Iambic_Keyer_Micro_V04_0 | Arduino 1.6.5
sketch_AB9A_Iambic_Keyer_Micro_V04_0
69 void loop() {
70   // put your main code here, to run repeatedly:
71
72   WPM = map(analogRead(0), 0, 1023, wpmMin, wpmMax);
73   T = 1200 / WPM; // linearizes the speed pot
74
75   if ((ditt) || (digitalRead(2))) {
76     if ((digitalRead(3)) && iambicB) { // Iambic B
77       dahh = true;
78     }
79     sendkey(1);
80     ditt = false;
81   }
82
83   if ((dahh) || (digitalRead(3))) {
84     if ((digitalRead(2)) && iambicB) { // Iambic B
85       ditt = true;
86     }
87     sendkey(3);
88     dahh = false;
89   }
90 }
91
92 Done Saving
Sketch uses 6,692 bytes (20%) of program storage space. Maximum is 32,256 bytes.
Global variables use 217 bytes (10%) of dynamic memory, leaving 1,831 bytes for local
variables. Maximum is 2,048 bytes.
```

Arduino Uno



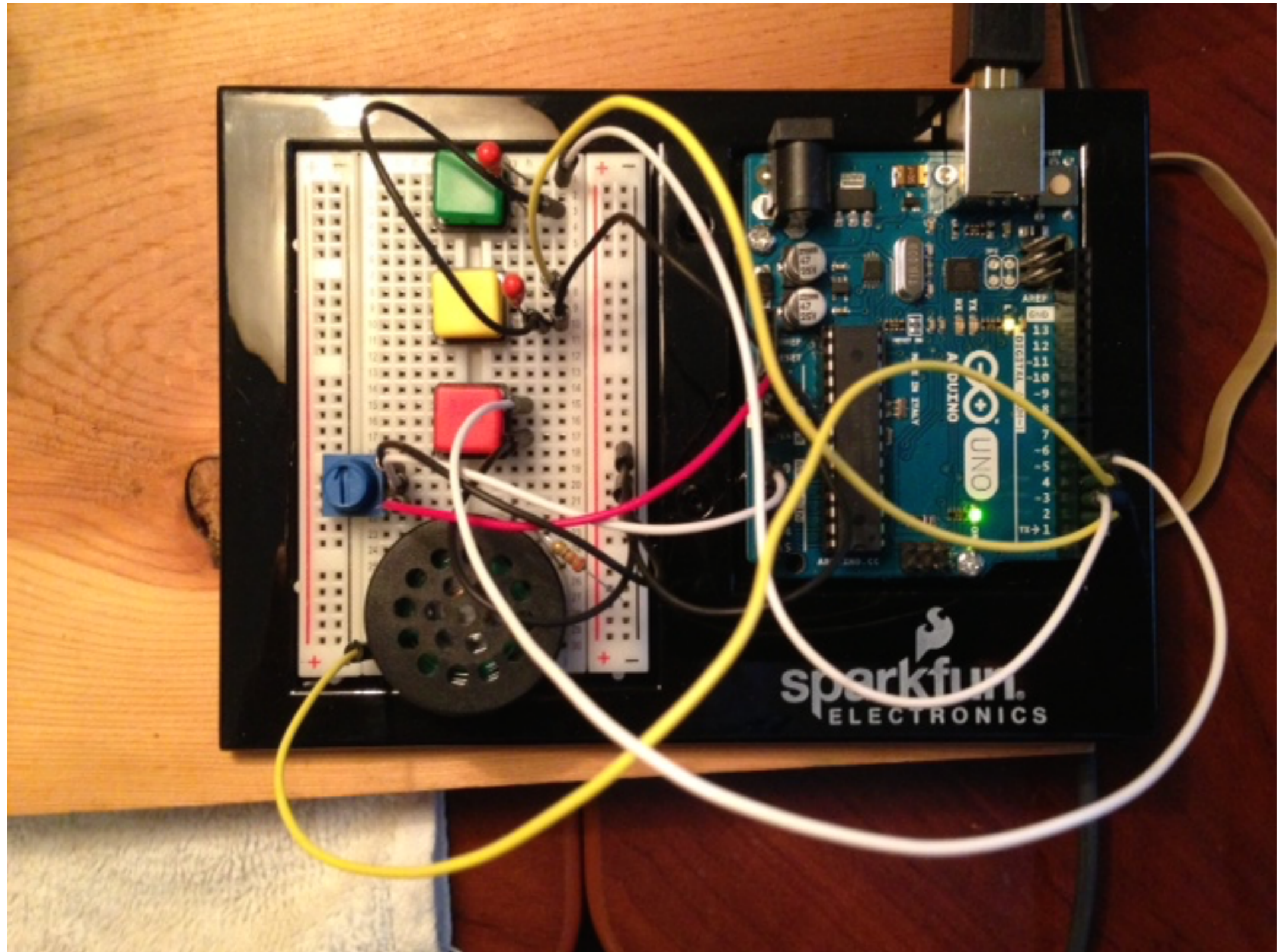
- Microcontroller
- 16 MHz
- 14 Digital Input / Output Pins
- 6 Analog Input Pins
- 32 KB Flash Memory
- 1 KB EEPROM
- 5 Volts



=



Prototype



Timing

- Morse Code is a combination of Dits and Dahs
- Timing of Code elements is important
- Dit time is the fundamental time reference
- Dah = 3 Dit times
- Inter-Element spacing = 1 Dit time
- Space between letters = 3 Dit times
- Space between words = 7 Dit times

Dit Time

Dit Time (in milliseconds) = $1200 / \text{WPM}$

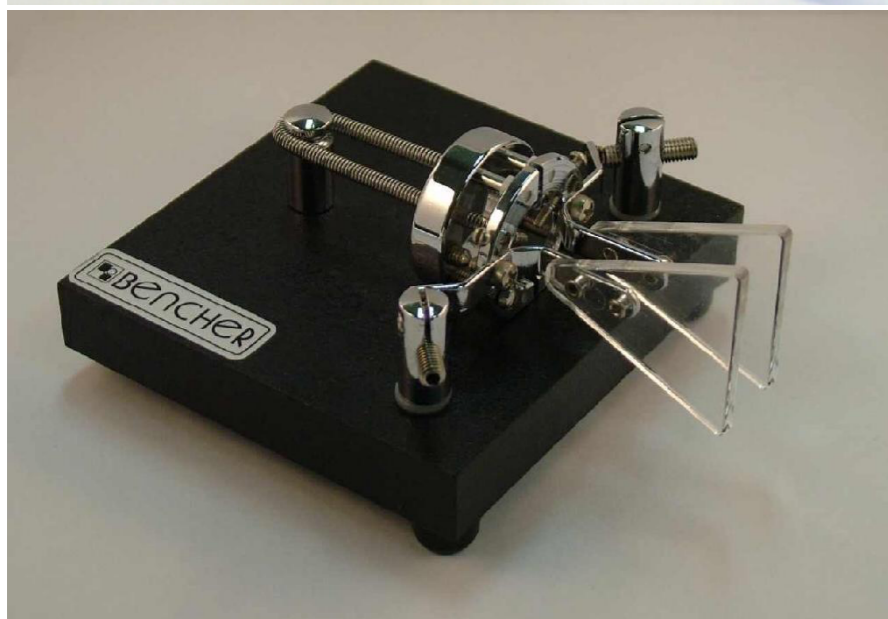


Straight Key

Keys
&
Paddles

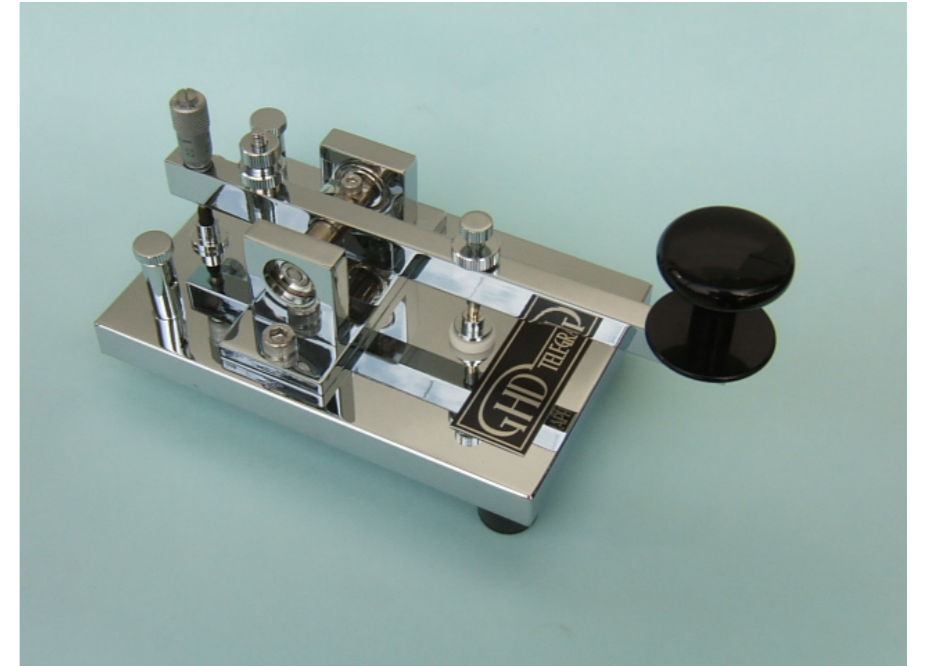


Single Lever Paddle



Iambic Paddle

Straight Key



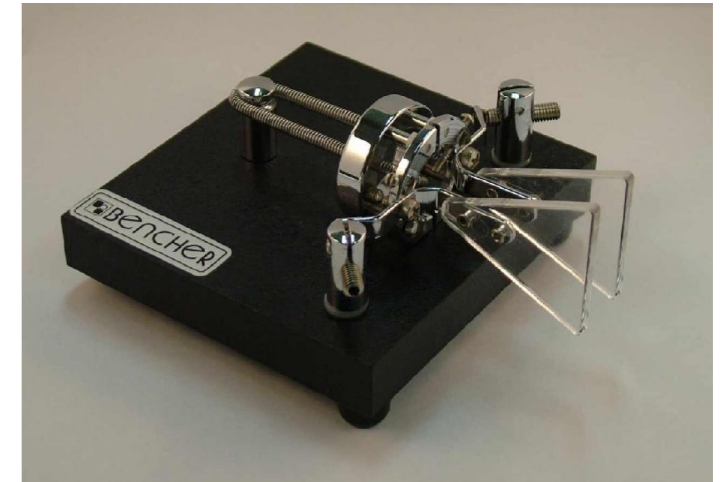
- Timing is determined by the skill of the operator
- As code speed increases, it becomes increasingly more challenging to maintain the correct timing relationships.
- Possibility of getting tendonitis or “glass arm”.

Keyer

- Requires a Paddle with 2 contacts, to know a dit from a dah.
- Accepts input from the paddle. And, sends precisely timed switch closures to the radio.
- Will repeat dits or dahs, as long as you hold the contact closed.
- Thumb ditters vs Finger ditters.



lambic Keyer

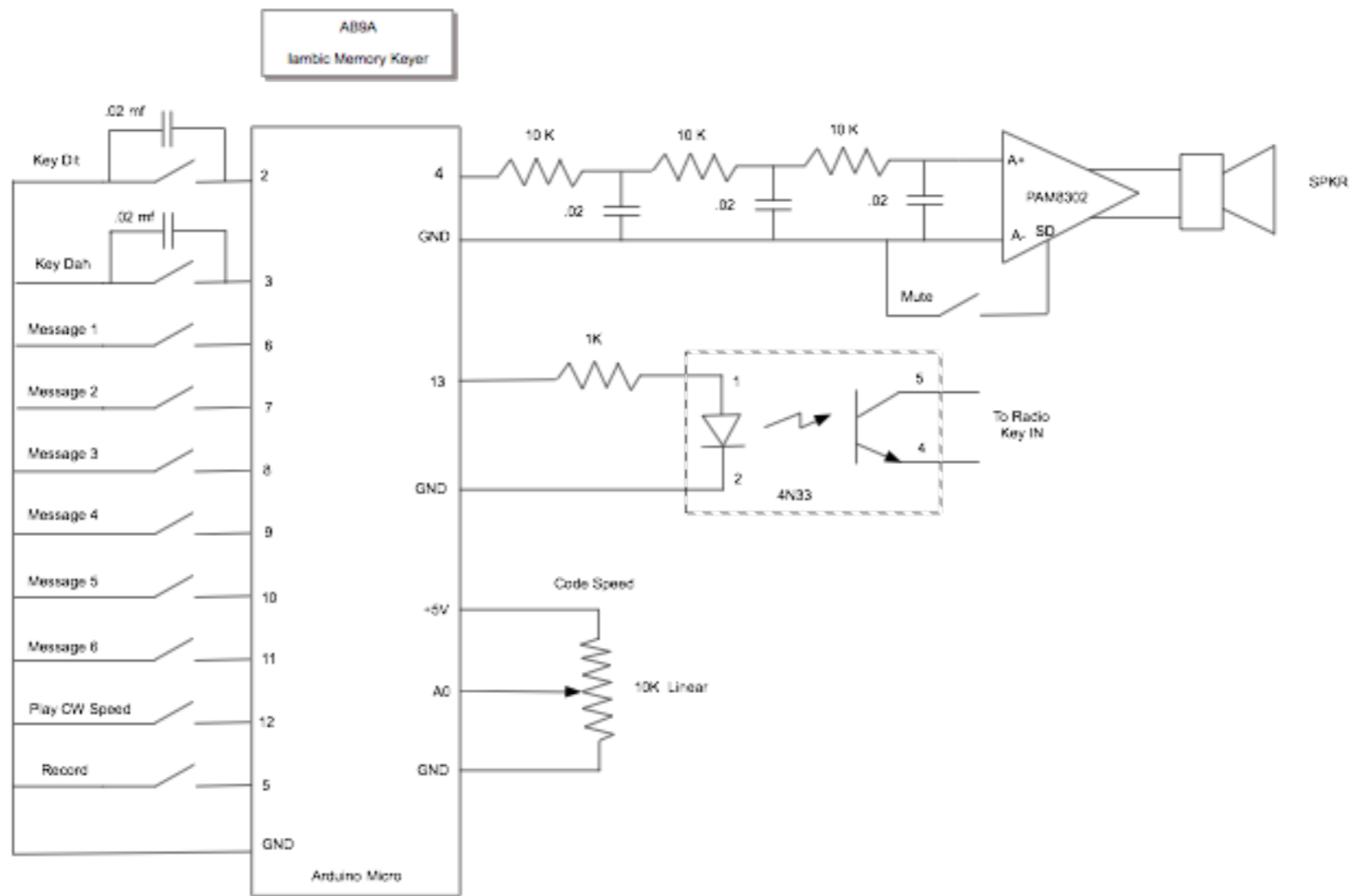


- Sometimes called a “squeeze keyer”.
- Requires a paddle with 2 levers & 2 contacts.
- Will send alternating dits and dahs, if you squeeze and hold both paddles.
- The letter C requires 1 squeeze vs 4 taps for a single lever paddle or straight key.
- Offers approximately 11% increase in efficiency.
- lambic A vs lambic B

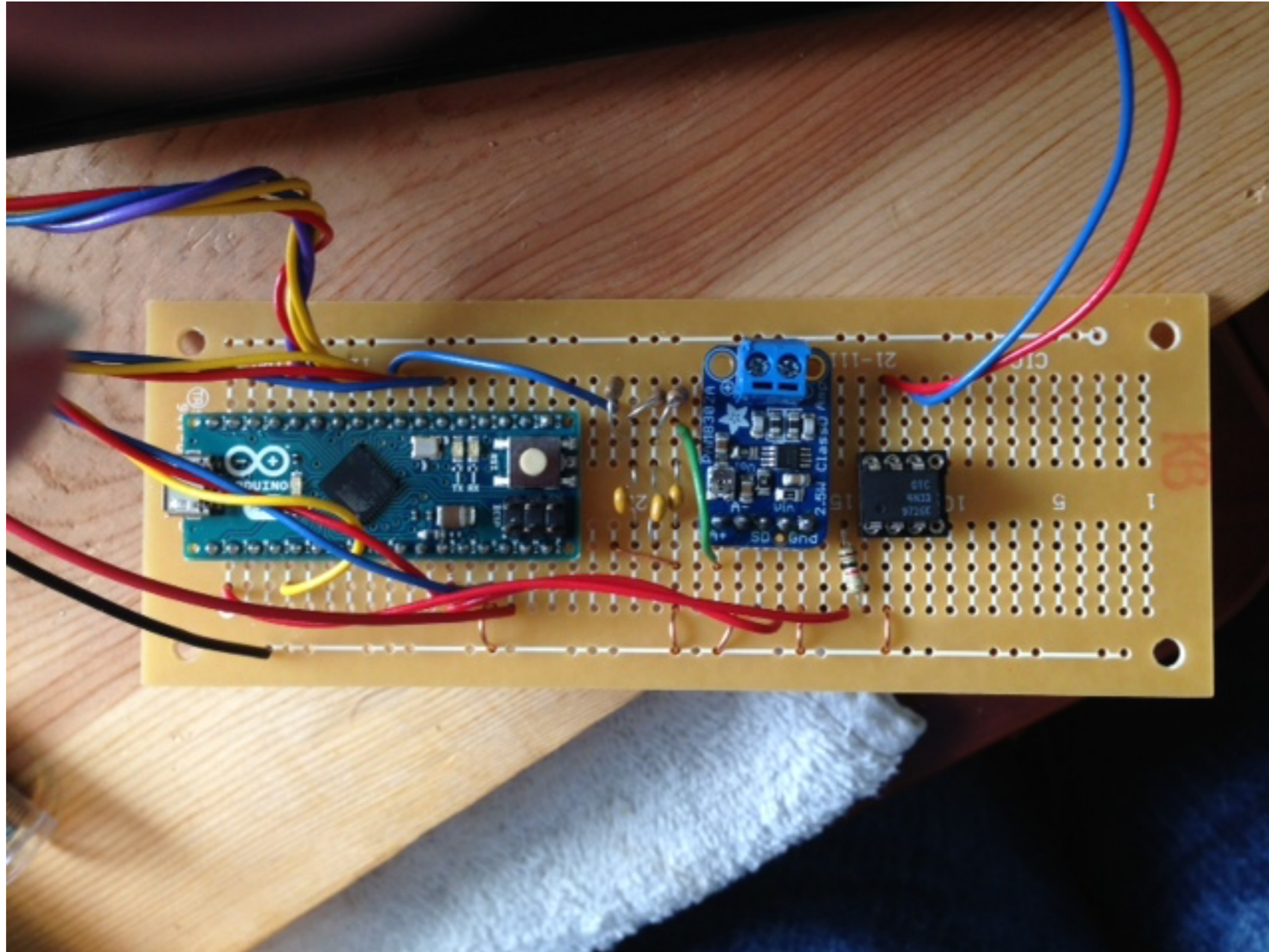
Memory Keyer



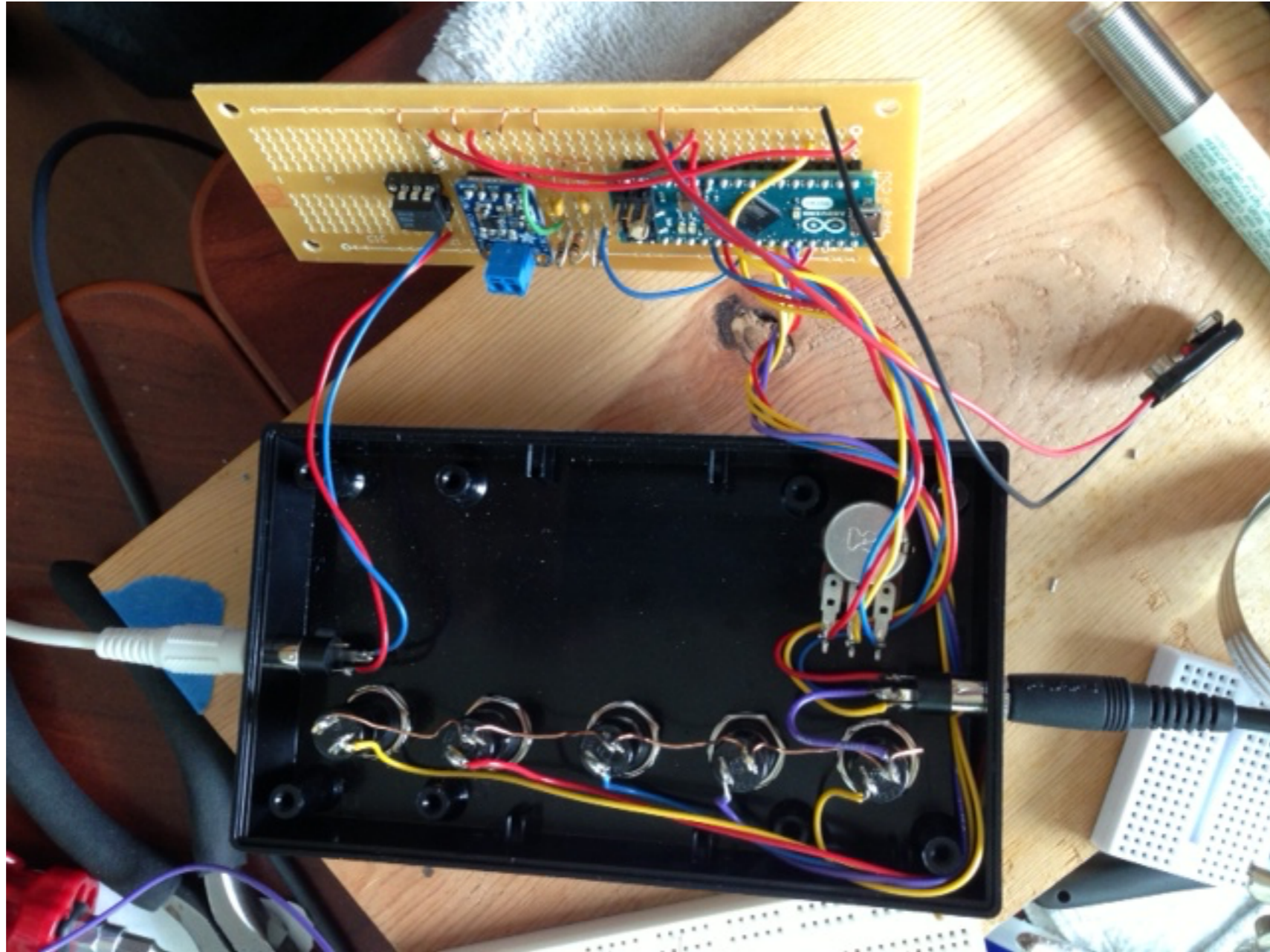
- Records messages, from the paddle or computer, and stores them in memory.
- Sends (Keys) a stored message to the radio when the appropriate switch is pressed.



Final Circuit Board

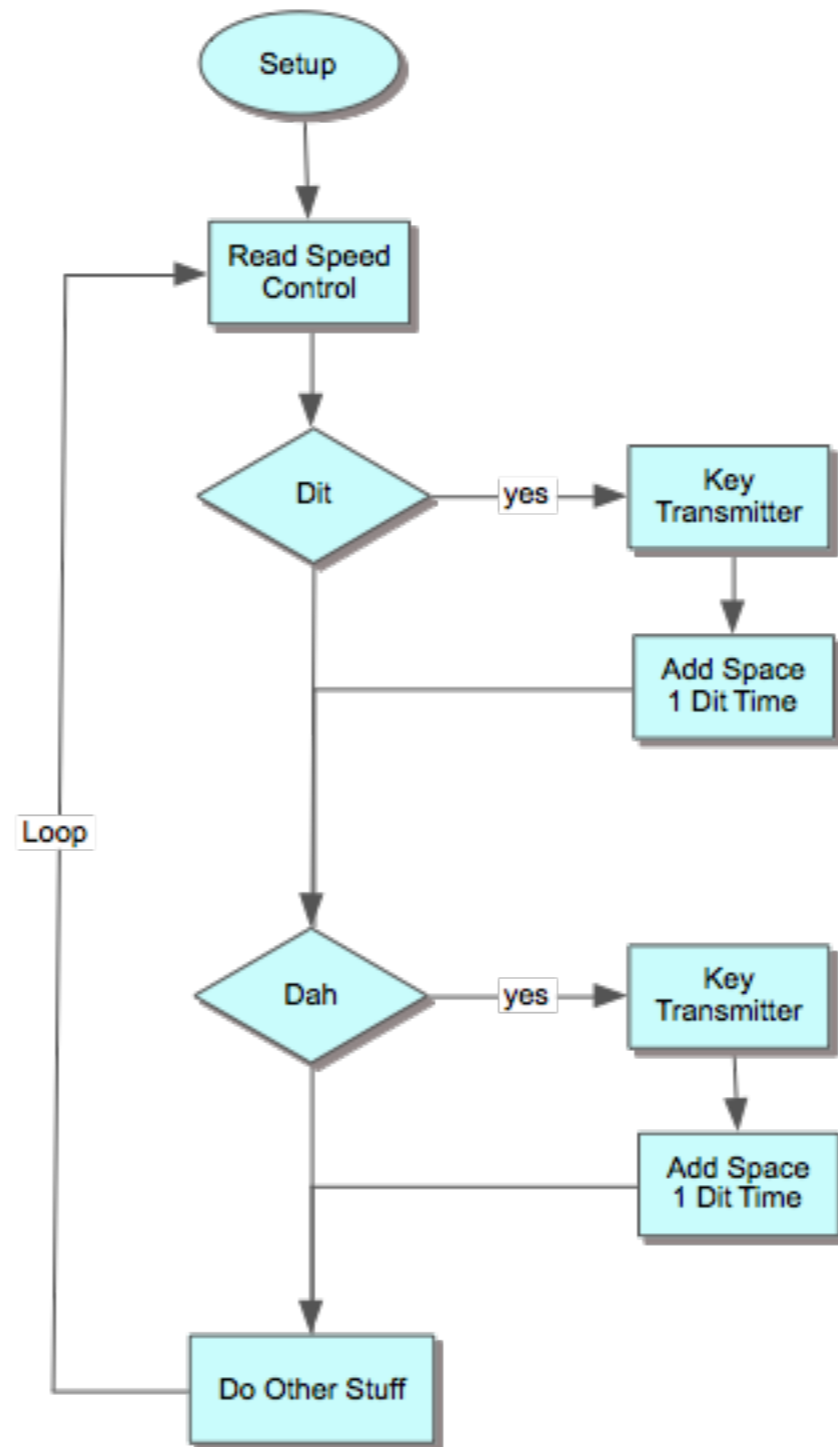


Inside Chassis Wiring



Final Assembly



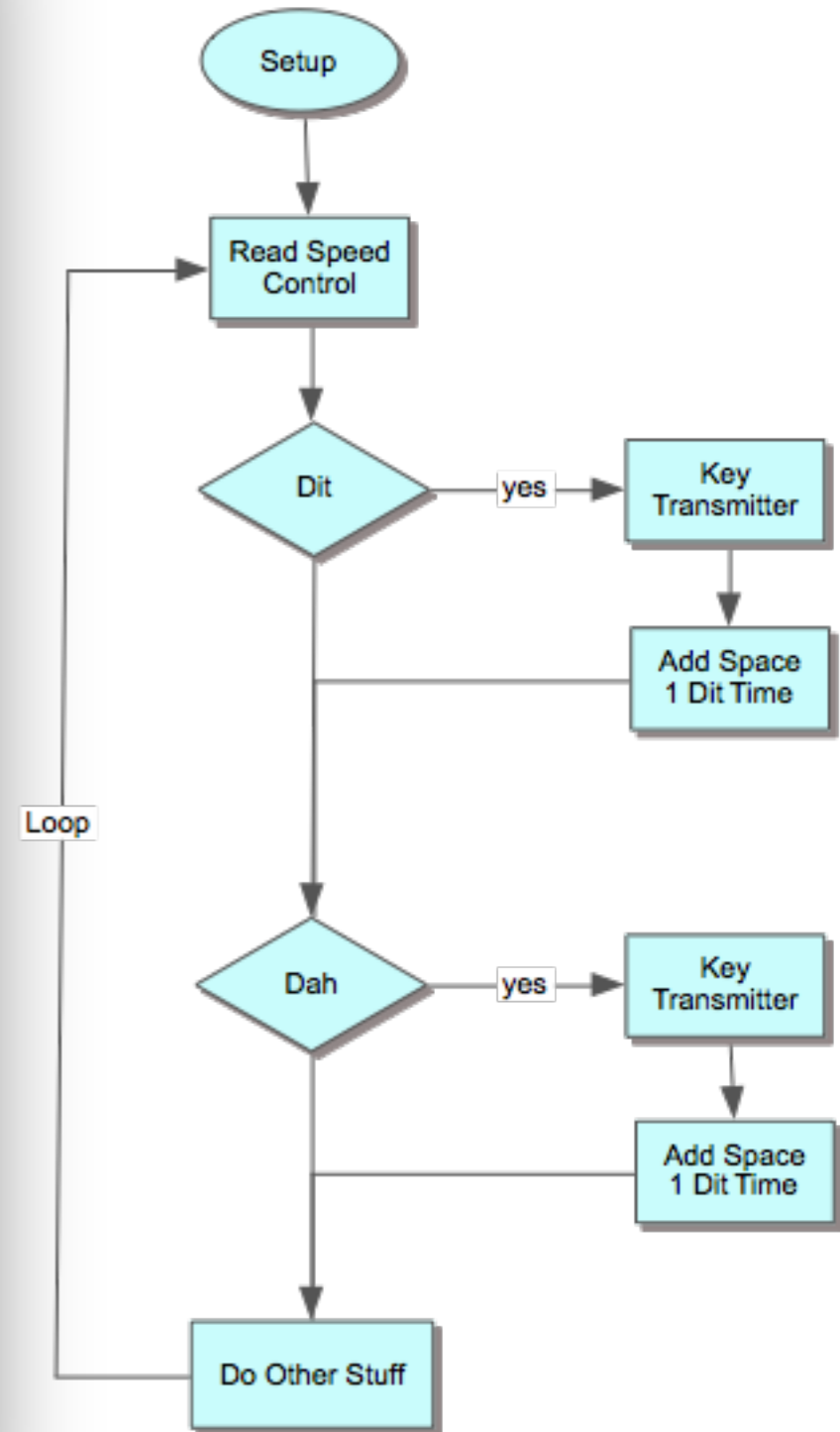


sketch_AB9A_lambic_Keyer_Uno_V04_0

```

42
43 void setup() {
44   // put your setup code here, to run once:
45   pinMode(2, INPUT_PULLUP); // Key dit
46   pinMode(3, INPUT_PULLUP); // Key dah
47   pinMode(5, INPUT_PULLUP); // Record Function Switch
48   pinMode(6, INPUT_PULLUP); // Message 0 Switch
49   pinMode(7, INPUT_PULLUP); // Message 1 Switch
50   pinMode(8, INPUT_PULLUP); // Message 2 Switch
51   pinMode(9, INPUT_PULLUP); // Message 3 Switch
52   pinMode(10, INPUT_PULLUP); // Message 4 Switch
53   pinMode(11, INPUT_PULLUP); // Message 5 Switch
54   pinMode(12, INPUT_PULLUP); // Play CW Speed
55   pinMode(13, OUTPUT); // OptoCoupler to key transmitter
56
57   digitalWrite(13, LOW); // Initialize Key output
58
59   attachInterrupt(1, dah, FALLING); // pin 3, reverse interrupt # for Micro ie. 0 to 1
60   attachInterrupt(0, dit, FALLING); // pin 2, reverse interrupt # for Micro ie. 1 to 0
61
62   //Serial.begin(9600);
63
64 } // ***** End of Setup *****
65

```



sketch_AB9A_lambic_Keyer_Micro_V04_0

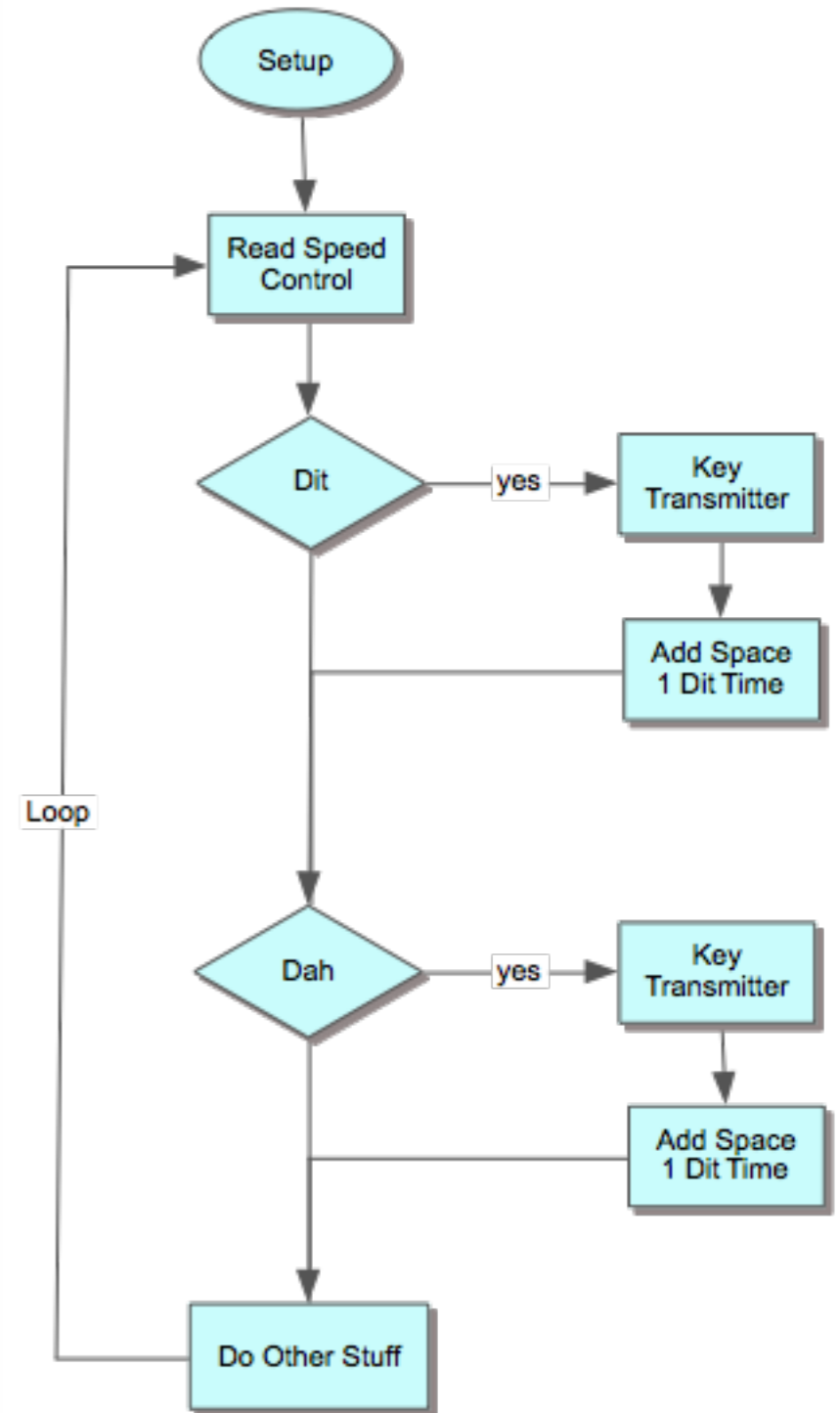
```

69 void loop() {
70   // put your main code here, to run repeatedly:
71
72   WPM = map(analogRead(0), 0, 1023, wpmMin, wpmMax);
73   T = 1200 / WPM; // linearizes the speed pot
74
75   if ((ditt) || (!digitalRead(2))) {
76
77
78
79
80
81     sendkey(1);
82     ditt = false;
83   }
84
85   if ((dahh) || (!digitalRead(3))) {
86
87
88
89
90
91     sendkey(3);
92     dahh = false;
93   }
94
95   }
96
97 }
98

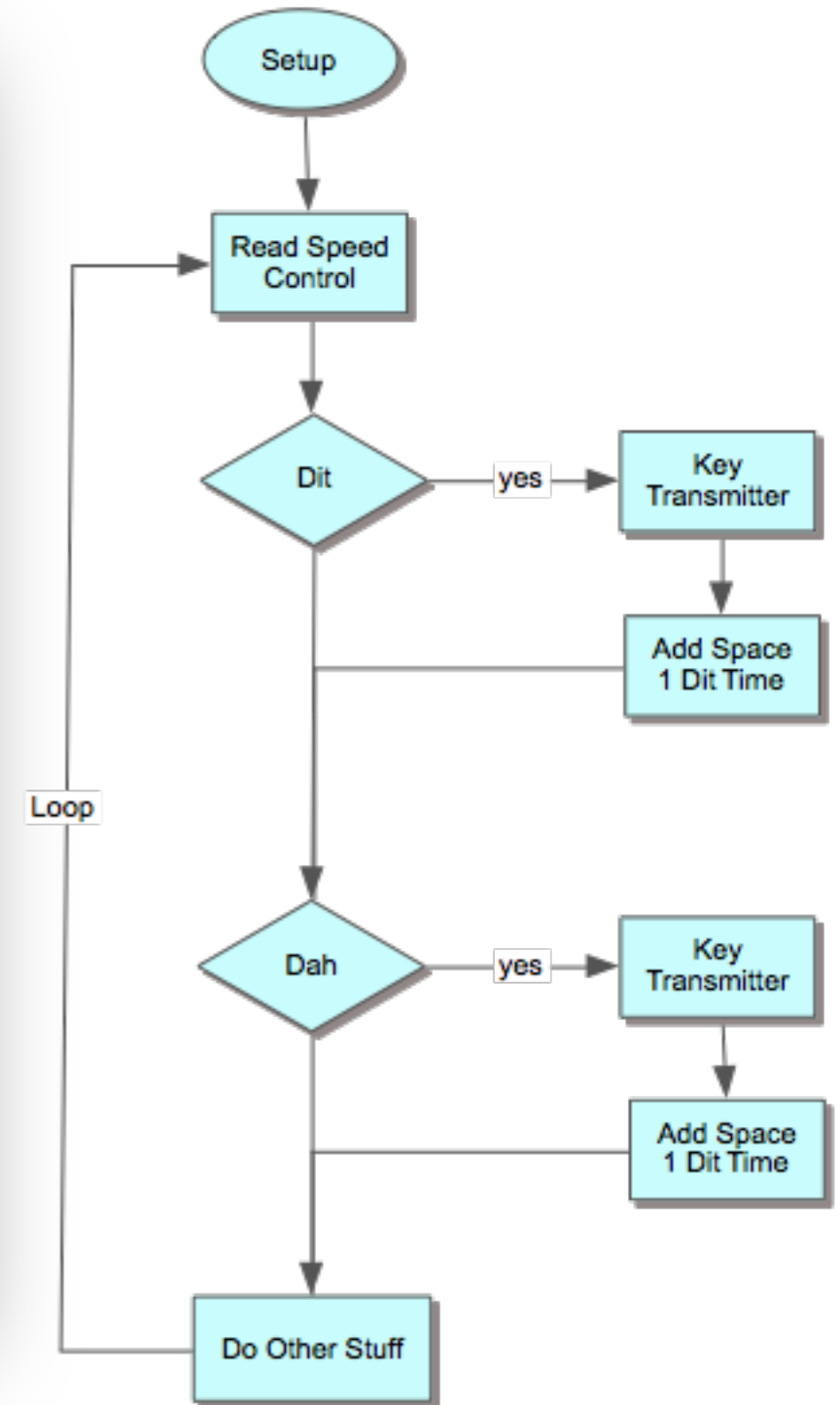
```

Done Saving.

Sketch uses 6,692 bytes (20%) of program storage space. Maximum is 32,256 bytes.
 Global variables use 217 bytes (10%) of dynamic memory, leaving 1,831 bytes for local variables. Maximum is 2,048 bytes.



```
sketch_AB9A_lambic_Keyer_Uno_V04_0 | Arduino 1.6.5
sketch_AB9A_lambic_Keyer_Uno_V04_0
548 }
549
550 // ***** Key Transmitter *****
551
552 void sendkey(int d){
553     // d is the number of dit times to delay
554
555     digitalWrite(13, HIGH); // Key transmitter on
556     tone(4, sidetone);      // side tone on, 900 hz
557     delay(d * T);          // dit times Key is on
558     noTone(4);              // side tone off
559     digitalWrite(13, LOW); // Key transmitter off
560     delay(T);               // insert one dit time between symbols
561
562 }
563
564
46 Arduino Uno on /dev/cu.usbmodemfa131
```



Features

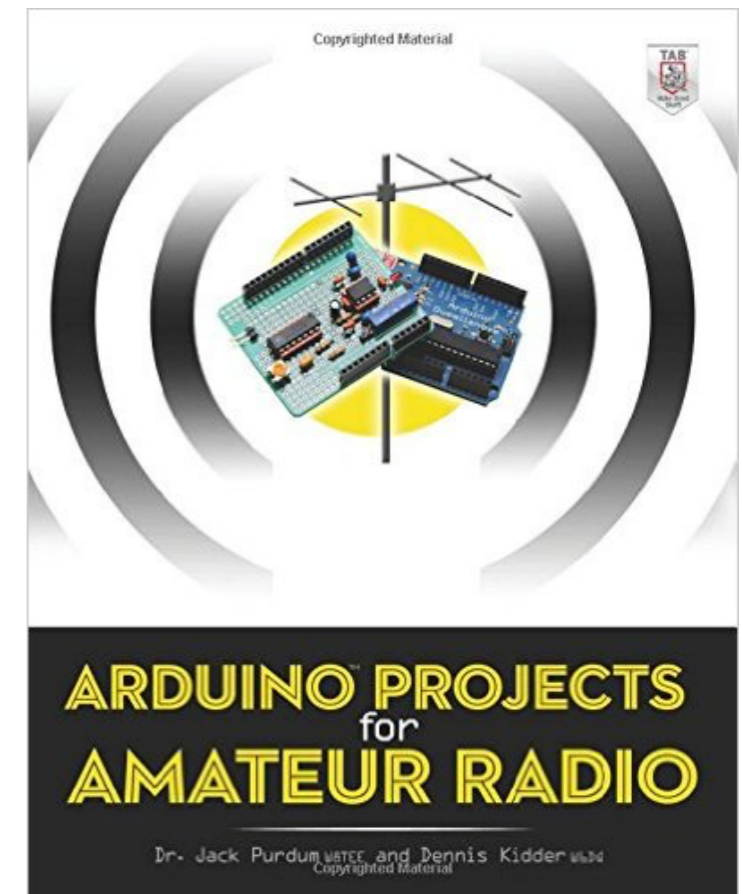
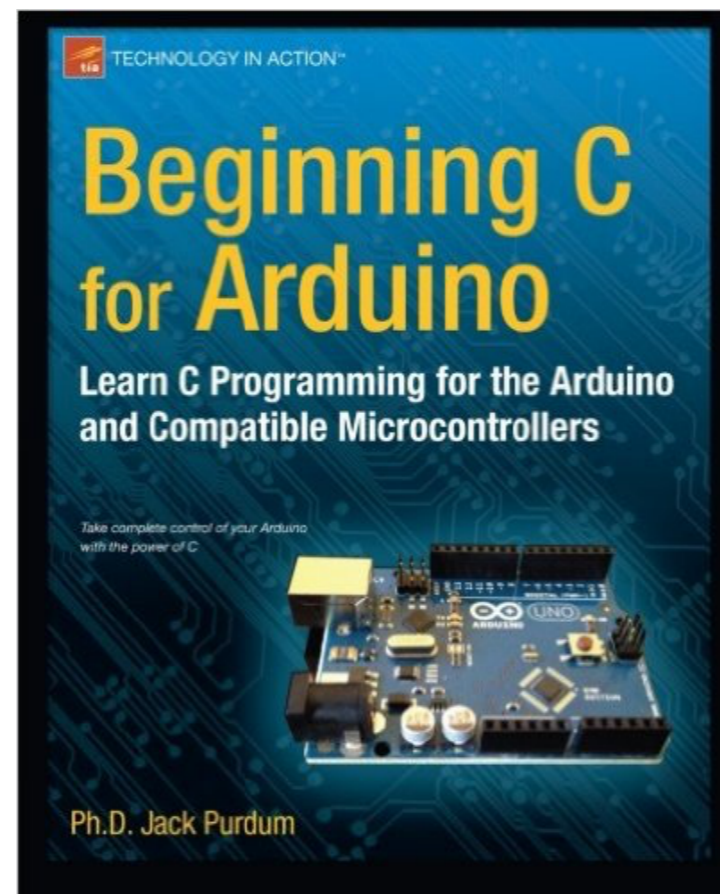
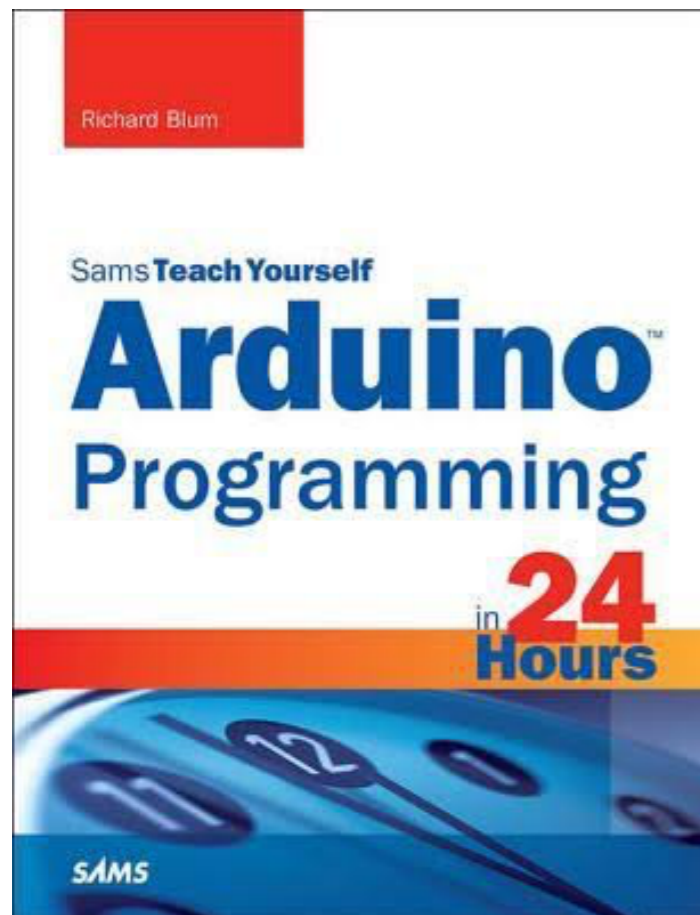
- No Menus
- Side Tone Speaker with Mute Switch
- Supports Iambic Modes A and B
- CW Speed Control
- Play CW Speed in Side Tone
- 4 Message Play Switches
- Record Messages with the Paddle
- Messages are stored in Non Volatile Memory
- Cancel Message Play with Paddle Tap
- Key Transmitter on to Tune.
- Optically Isolated Output to Key the Radio
- Code Practice Oscillator



www.arduino.cc

www.adafruit.com

www.sparkfun.com



Next Project

Ultra Wideband VFO

