The Morse Machine
- for Morse code demonstration, learning and practice

Torbjørn Skauli, LA4ZCA (tskauli@gmail.com)

version 2018-12-20

Introduction

Morse code may be old fashioned, but it still fascinates people, old and young. This text describes a setup for Morse code practice and demonstration. The setup can be built easily and at very low cost using 3D printed parts and readily available components. The Morse Machine can be built with both straight key and paddles, for example as in in Figure 1 below. The Morse Machine can obviously be used for keying a transmitter, but the intended purpose has been to demonstrate or teach Morse code, and not least to promote amateur radio. Thanks to LA3ZA for the idea.

The functionality is based on the K3NG keyer code for an Arduino microcontroller, configured with an LCD display and keypad module. Sent characters are played as tones in a speaker, and also shown decoded in the display. The speed and volume are adjustable. Power can be supplied as 12V DC or through USB. The leftmost Enter button on the keypad enters command mode, where some transmit practice functions can be enabled, see documentation for the K3NG keyer. The rightmost button does a full Reset. Otherwise, the buttons are not used in the current version. In a future upgrade, the features of the Morse Machine can be extended by adapting the software, including Morse code reception practice. Help is wanted for extending the code, see below.

A Morse Machine can be constructed in many different forms, depending on the preferences of the builder and on the parts available. This guide assumes some basic familiarity with building electronics, such as how to solder and mount the various parts, and how to interpret the schematic to get the connections right. With such basic skills, the construction is straightforward.

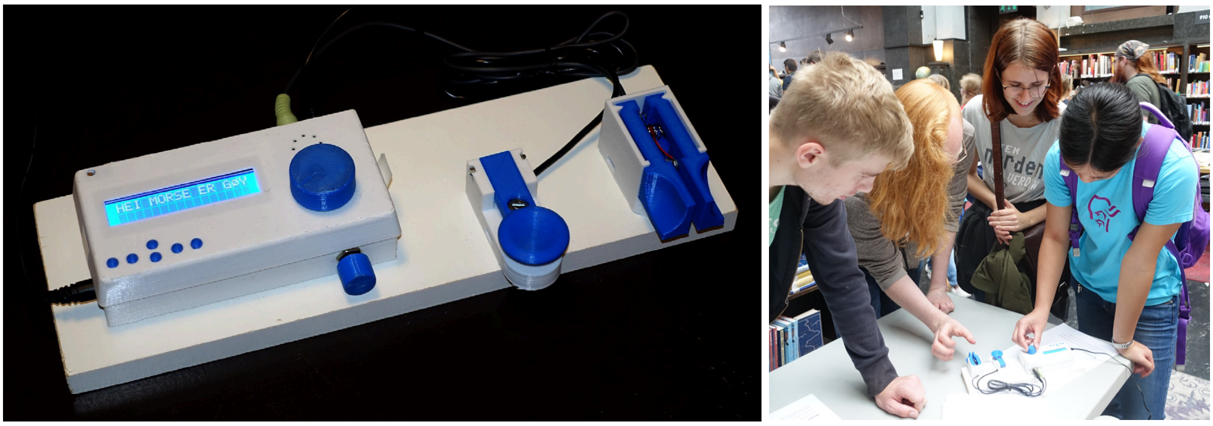


Figure 1. The Morse Machine. Left: Setup with key and paddles on a wooden board. Right: Interested visitors at an exhibit in Oslo main library see their Morse characters appear on the LCD display.

Materials

The Morse machine is built from readily available parts:

- Arduino Uno microcontroller board. Also download and install the Arduino software.
- LCD display module ("shield") which plugs directly into the Arduino, of the type shown in Figures 2 and 4. For more information, see for example here.
- Arduino code from GitHub, see below.
- Potentiometer for speed control, 1 to 20 kΩ, linear.
- Potentiometer for volume control, 1 to 100 kΩ, of the "logarithmic" type normally used for volume controls. (Alternatively, use a linear potentiometer and connect a resistor from the wiper/center to ground with a value of 1/4 to 1/5 of the potentiometer resistance.)
- Knobs for the potentiometers, for example based on the customizable design here.
- 2x 3.5 mm jacks for key and paddles.
- Audio amplifier module such as this one, see discussion below.
- Speaker: Any type will do, but a rectangular type with dimensions 28x40 mm fits well in the 3D-printed cabinet. Such speakers are widely available, for example here or here.
- 3x 10 nF capacitors
- 3D printed cabinet based on the STL files or OpenSCAD source provided with this document, modified from this parametric design on Thingiverse.
- Straight key, for example the 3D-printed design here, shown in Figure 1.
- Paddles, for example the 3D-printed design here, shown in Figure 1.
- Mounting base, for example a piece of painted board, as shown in Figure 1.
- A few additional parts, as shown on the schematic in Figure 3, if you want to use the Morse machine for keying a transmitter.

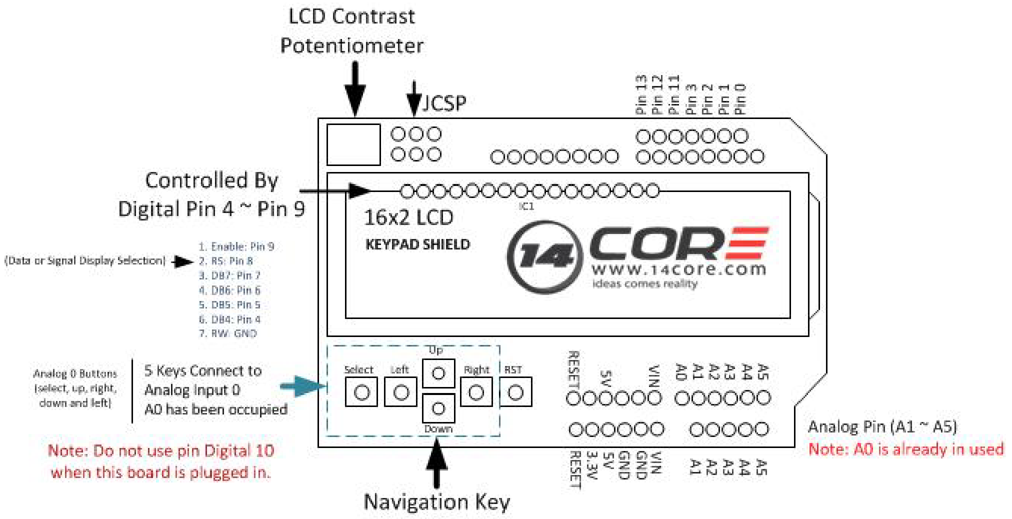


Figure 2. Pinout of the Arduino LCD shield used for the Morse Machine. (from 14core.com CC-BY-SA)

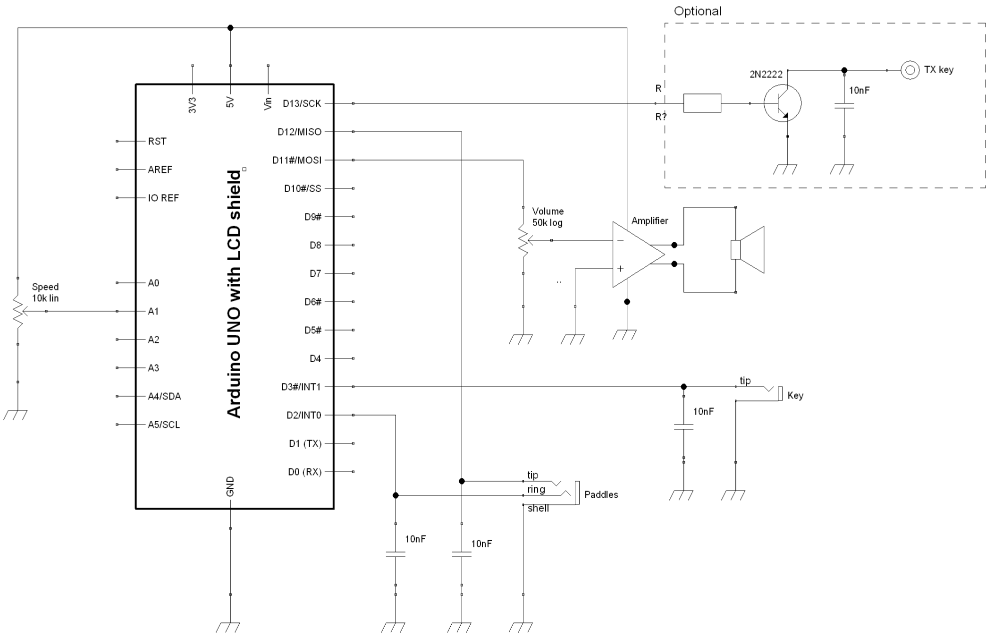


Figure 3. Schematic of the Morse Machine, based on the K3NG keyer. The keying output is not needed if the unit is to be used for exhibits and practice. Here, one channel of a stereo audio amplifier module is used for the speaker output. An alternate speaker output is shown in the original K3NG schematic.

Circuit description

The schematic of the Morse Machine is shown i Figure 3. This is essentially a simple version of the suggested schematic for the K3NG keyer. The keyer output is optional if the unit is going to be used for demonstration and practice. In the build shown in the pictures here, the speaker output makes use of one channel of a cheap audio amplifie module, but it is also possible to use a simple transistor to drive the speaker. As shown, the key and paddles are connected via 3.5 mm jacks, but in a permanent setup it is obviously possible to simplify the build by connecting wires directly. Do not omit the decoupling capacitors if a transmitter is nearby, because otherwise the RF will key the arduino randomly.

The LCD module uses several of the I/O pins of the Arduino for driving the display, light and buttons. The rest of the Arduino pins are routed to pads at the edges of the LCD module, as shown in Figure 2. From there, it is easy to connect wires to the rest of the circuit. The version of the code posted here is configured for the connections shown in the schematic. There are a few different variants of the LCD "shield" module on the market, so some care is needed to get the connections right. The K3NG code can be modified according to the choice of I/O ports. Avoid Arduino ports 0 and 1, which are used for communication with the PC.

Comments on the code and desirable future upgrades

The code is based on the original K3NG code, with some adaptations. It resides on Github here. Begin by downloading the code from GitHub as a .ZIP file, and unzip to a local directory. Copy or move the content of the "Libraries" subfolder to the "Libraries" subfolder of the Arduino software, typically "C:\Program Files\Arduino\Libraries". Open the file "Morse_Machine_K3NG_keyer.ino" in the Arduino software, and compile/upload to the Arduino Uno board. There is lots of help on the internet if you are unsure how to do this, start with arduino.cc and the K3NG keyer site.

The keyer software has lots of features and options that can be changed, selected and deselected in the source code. The changes that have been made to the original code can be found by searching the code for comments containing "LA4ZCA". The main changes are:

- enabling required features in the file "keyer_features_and_options.h", including enabling non-English character support (ÜÆØÅ etc.).
- setup of hardware parameters in the file "keyer_hardware.h"
- some changes to the default configuration in the file "keyer_settings.h"
- change of the welcome message
- EEPROM reset every time the Arduino resets, to avoid accidental config changes
- bugfixes for reading the LCD shield keypad and handling non-English character echo from the straight key

More work on the software is desirable to enable some very nice features of the keyer software that are currently not accessible: There are many built-in practice modes, but they can only be activated by enabling the command line interface to a computer (which will not fit in the Uno memory space). It would be great if someone would implement these changes:

- a menu system for accessing training and practice modes using the LCD and buttons
- displaying letters on the LCD during code practice

Construction

The Morse Machine can be constructed in many ways according to your preference, using any suitable cabinet and a variety of parts. If you have a 3D printer available, and obtain parts with suitable dimensions, you may find it practical to build the Morse Machine as illustrated in Figures 4 to 7 using the 3D models listed in the parts list.

Figure 4 shows wires attached to the Arduino connections on the LCD shield according to the schematic in Figure 3 (without the optional TX output). Figures 5 and 6 show the mounting of other parts in the side compartment of the 3D printed enclosure. The enclosure as shown is adapted to the rectantgular speaker, mounted with glue, and other parts in use here but the OpenSCAD source file can be modified to accommodate other part dimensions. As shown in Figure 6, the speaker amplifier module can be simply mounted by double sided tape to the back of a potentiometer or the speaker. Figure 7 shows the completed interior of the Morse Machine electronics.

The enclosure is designed to have the top part snap in place on the bottom part. In practice, the fit may not be good enough. Usually, the LCD module has some solder joints that rest on the Arduino power connector so that the enclosure does not close properly. These solder joints may need to be trimmed down. Still the enclosure may not close properly. I have used pieces of adhesive tape from the top to the bottom on each side to hold the enclosure together.

If you use the 3D printed key and paddles then they will need to be mounted to a good support, normally by screws fixing the base parts onto a sturdy object such as the piece of painted board seen in Figure 1. You may want to modify the code to have screw holes in the Arduino cabinet too, or you may mount it the way I did, with double-sided tape.

I hope the Morse Machine will be an enjoyable project for you and for the other people that get to play with it.

73 de Torbjorn, LA4ZCA (tskauli@gmail.com)

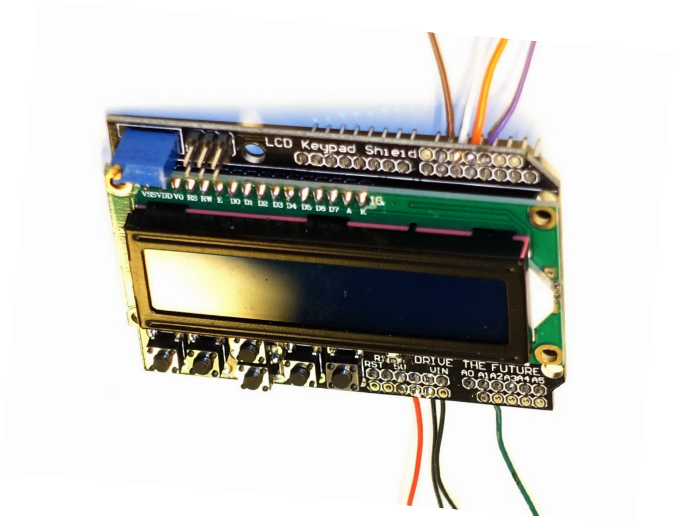


Figure 4. The LCD shield with wires attached.

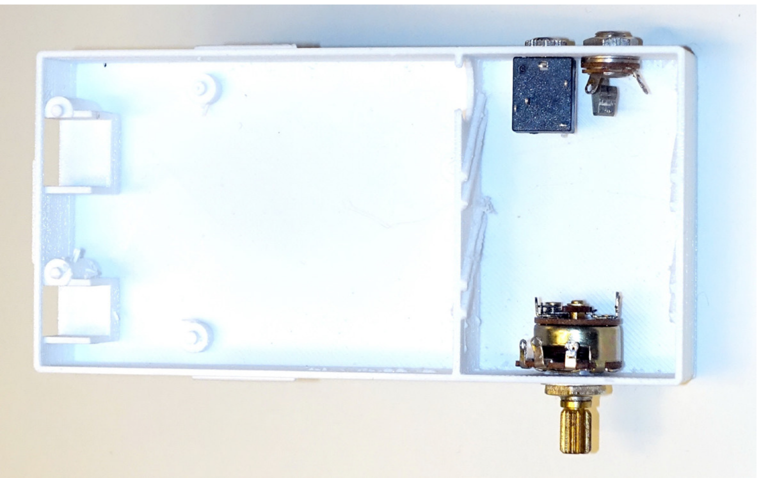


Figure 5. Bottom part of the cabinet with volume control potentiometer and jacks for the paddles and straight key. The two jacks are for 3-poled (left) and 2-poled (right) 3.5 mm plugs, hence the difference.

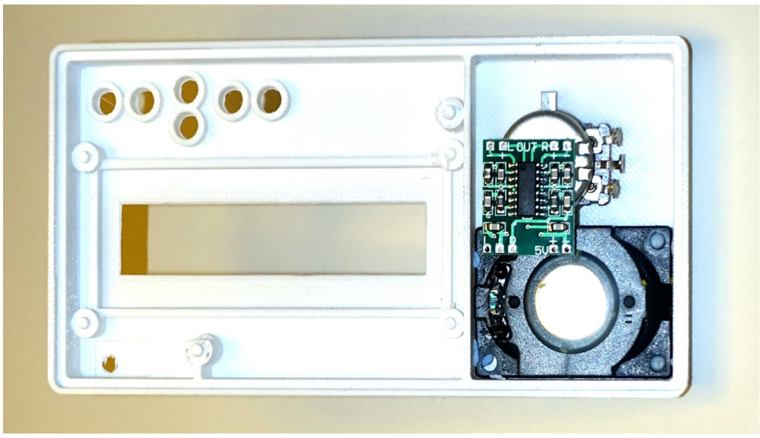


Figure 6. Top part of the cabinet with the speaker, glued in place, and speed control potentiometer. The amplifier module is mounted on the volume control potentiometer by double sided tape.

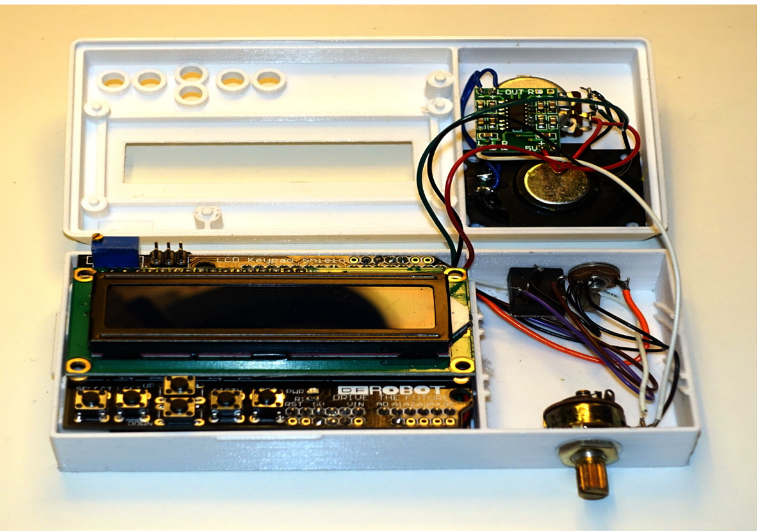
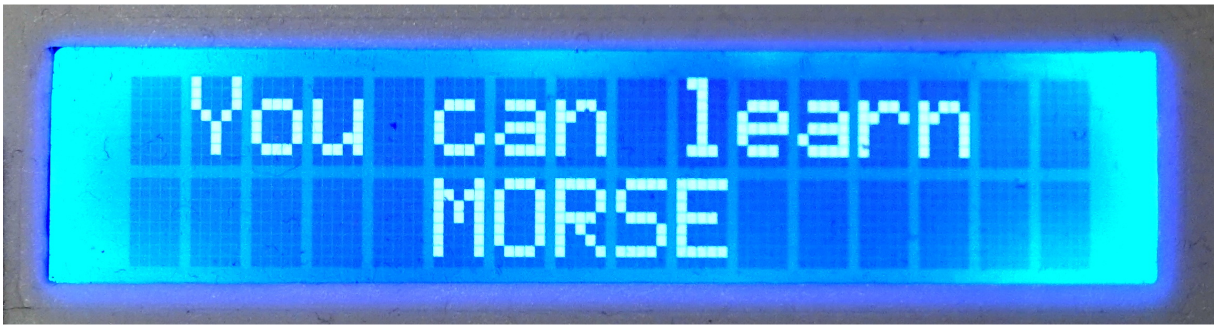


Figure 7. The completed electronics for the Morse Machine. (Decoupling capacitors are missing in the picture, they can be soldered across the jack pins.)



```
 You can learn
    MORSE
```

Morse Machine opening message, accompanied by MORSE sent in code.