1 INTRODUCTION

There's a python program that will allow to transform any set of alphanumeric characters into a set of Morse signals from a GUI. With the help of an arduino and a circuit, the Morse signals are then showned through a speaker, whose volume is controled by a potentiometer, and a LED.

2 MORSE CODE

Morse code is a character encoding scheme used in telecommunication that encodes text characters as standardized sequences of two different signal durations called dots and dashes. On Figure[1] you can find the correct correspondence. This program runs with a time unit of 1 second.

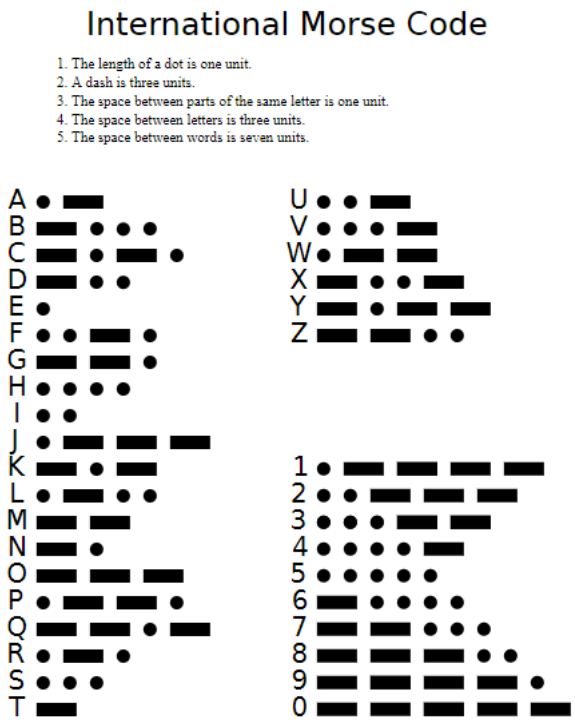

Figure 1: Morse Code Table

3 PYTHON CODE

3.1 GUI

The user interface is a pop-up window, with a background image of the portuguese navy, and two inputs: "stream" - the string the user wants to code, and "com" - the com port that the user wants use to communicate to their arduino.

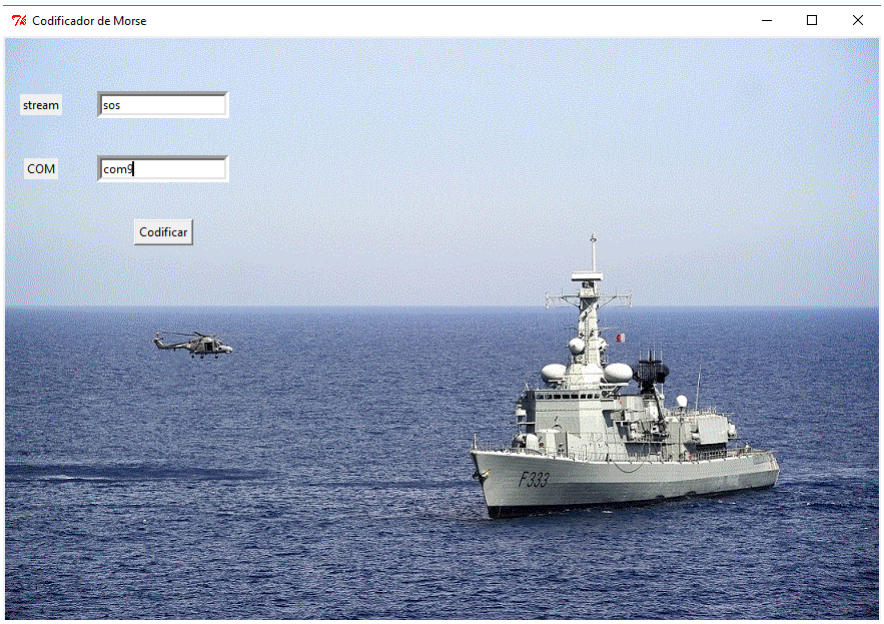

Figure 2: Graphic User Interface

The inputs then are processed by a function gone by the name of process().

Note that the word "code" is used here as a synonym of (0, 1, 3, 5) which corresponds to (idle time, dash, dot, space).

3.2 Processing user input

In the process() function, the Alpha and Morse lists, contain which the alphanumeric elements and each element code respectively. In this sense, len(Morse)=len(Alpha). Comparing the values inserted by the user with the Alpha list, the same index

of each match is used to search for the respective code in the Morse list. The output is then a list where each element is in code. Which, in turn, is sent over to the "data" variable in the .ino file, via serial.

4 ARDUINO

The .ino code has to be inside a folder with the same name. The delay() function defines each code as a multiple of 1 = 1milisecond. The data is read and writen as output to digital pin 10 and 11, which I'll send 5 Volts if data reads dash(1) or dot(3).

5 PROJECTING THE CIRCUIT

After the circuit schematic were designed using: EAGLE 5.0.0 - to design the schematic and pcb boards 5Spice - to simulate said circuit While designing the PCB, most of the wires were replaced with the lower layer of the board that has fabricated runways. This way we reduce the amount of necessary wires in the upper layer. There was a small conection between two lines next to each other in the upper layer, so we short-circuited it with the alloy in the lower layer.

The Arduino signal sends a maximum voltage of 5V, which, with a small resistance is good enough to power the LED. In the speaker circuit a power transistor alows to increase the current in the speaker branch. As for the potentiometer, it works as a volume controller.

In figure[3] is represented a schematic of the circuit. Or rather, the two independent circuits.

- R1 = 1kohms
- R2 = 1kohms
- R3 = 100ohms
- Q1 = 2SC1306
- D1 = 5mm, your favorite colour
- SP1 = any speaker with low resistance

In figure[3] T1 = Q1, from bottom to top we have the base, collector and emitter.

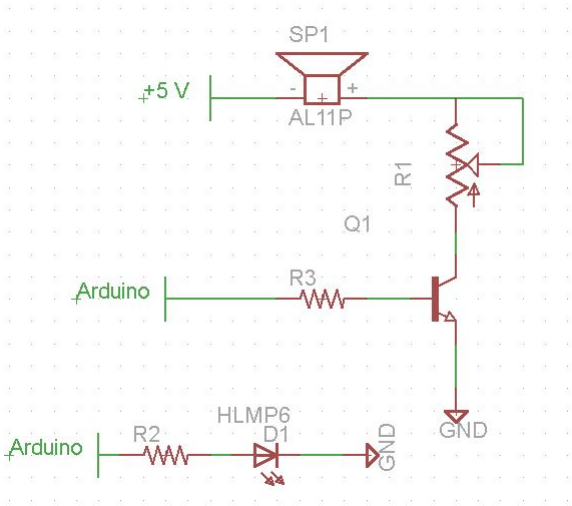

Figure 3: Diagram

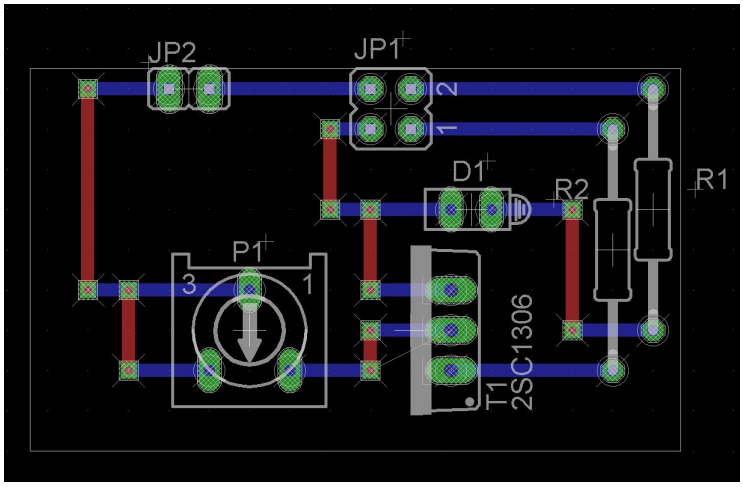

Figure 4: PCB

Since we are dealing with square waves, all that can be output to the speaker are pulses.

6 HOW TO ASSEMBLY

On figure[4] it can be seen two jumpers:

- JP2 is where the user connects the ends of the speaker;
- JP1:
  - Top Left: 5V from the Arduino
  - Top Right: Arduino pin 10
  - Bottom Left: GND form Arduino
  - Bottom Right: Arduino pin 11

After this simple assembly, as soon as you run the program you're good to go!