RadCom - Morse Tutor

Author: M.D. Waller

Version 1.0.0

**Contents**

[1. Morse Tutor 3](#_Toc378767216)

[2. What makes a good Morse Tutor? 3](#_Toc378767217)

[3. The Design 3](#_Toc378767218)

[3.1. The Oscillator 4](#_Toc378767219)

[3.2. The Controller 4](#_Toc378767220)

[4. Oscillator 4](#_Toc378767221)

[5. Processor Unit 5](#_Toc378767222)

[6. Vero board Layout 6](#_Toc378767223)

[7. List of parts 10](#_Toc378767224)

[7.3. Regulated Power Supply 10](#_Toc378767225)

[7.4. Oscillator 10](#_Toc378767226)

[7.5. Processor Unit 11](#_Toc378767227)

[7.6. Miscellaneous 11](#_Toc378767228)

[8. Possible enhancements 11](#_Toc378767229)

[9. References 11](#_Toc378767230)

# Morse Tutor

Learning Morse code is never going to be easy. Ultimately it comes down to immersing yourself in the sound until it becomes second nature. There are various ways of doing this such as using pre-recorded tapes, CD’s, PC Software or using one of the many on-line web sites. For me there is nothing better than having something portable that I can take with and turn on as and when I get the opportunity.

“Get hold of portable property.” Mr. Wemmick – Great Expectations by Charles Dickens.

Many of you may remember the Datong D70 Morse Tutor from the early 1980’s. I had one at the time and found it very useful; indeed without it I would probably have not passed the Morse test. This project is to build a similar device with extra functionality based around the new technologies available today.



# What makes a good Morse Tutor?

To start with it must cover the Morse alphabet, letters, numbers and punctuation and allow for any combination of the three to be sent. The original D70 did not support punctuation as the Morse test did not require you to know it. It must allow the speed of the Morse to be changed and the spacing between characters to be changed to provide extra thinking space while learning. Ideally it will also allow the tone and the volume to be varied too.

# The Design

The design is split into two distinct parts, the first being the oscillator and the second being the controller.

## The Oscillator

The oscillator had to provide both tone control and volume control. Searching the internet turned up a Morse oscillator designed by “Kent Engineers” based on the CMOS 4047b astable multivibrator. Permission was duly sought and granted to utilize this in the project.

## The Controller

The purpose of the controller is to generate five figure groups of random characters abiding by the chosen speed, delay and selected character groups. There are many micro controllers on the market to choose from but I had in stock a number of PICAXE 18M2 processors so the decision was made for me.

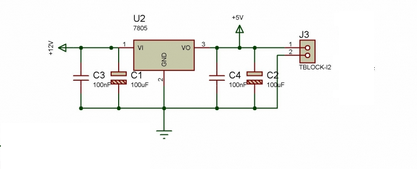
The PICAXE range of processors are both easy to interface to and easy to program. They are programmed using BASIC and free development tools are available on the internet. The development tools run on most platforms and talk to the PICAXE via a serial cable connected to a couple of the PICAXE pins and ground. The average PICAXE can support digital and analogue input and many of the standard serial interfaces.

The speed and delay controls are simple potentiometers which are read by the PICAXE internal analogue to digitals converters and turned into a binary representation which is then honoured as the characters are generated.

Character group selection is done via three switches, one to control whether alphabetic characters ‘a’ through ‘z’ are generated, one to control whether digits ‘0’ through ‘9’ are generated and one to control whether punctuation characters are generated. Any combination of the switches is allowed to generate characters tailored to your learning requirements. If all three switches are off then this is the same as all three switches being on and characters are generated across all three groups.

The design is based on the PICAXE 18M2 processor. I could have chosen one of the smaller offering but the 18M2 was already in stock.

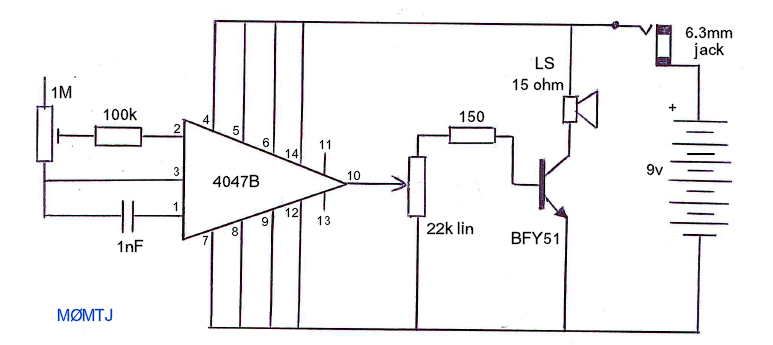
One aim of the project was to make the device portable. To this end it must run off battery but also have the ability to be driven from an external power supply. The PICAXE processor requires a supply voltage of 5 volts so a simple regulator circuit based on a 7805 will be used. A typical circuit looks like:



This is just a typical circuit, looking around the internet there are many variations on this theme some with capacitors some without!

# Oscillator

I wanted an oscillator that provided control over both volume and tone. Looking around the internet I found this:



See: <http://www.mds975.co.uk/Content/amateur_radio_projects.html>

# Processor Unit

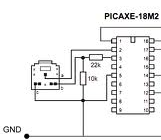
The processor unit is based on the PICAXE 18M2

On reflection this could all be done on a smaller variation of the PICAXE processors but I didn’t have them in stock!

The pins being used are:

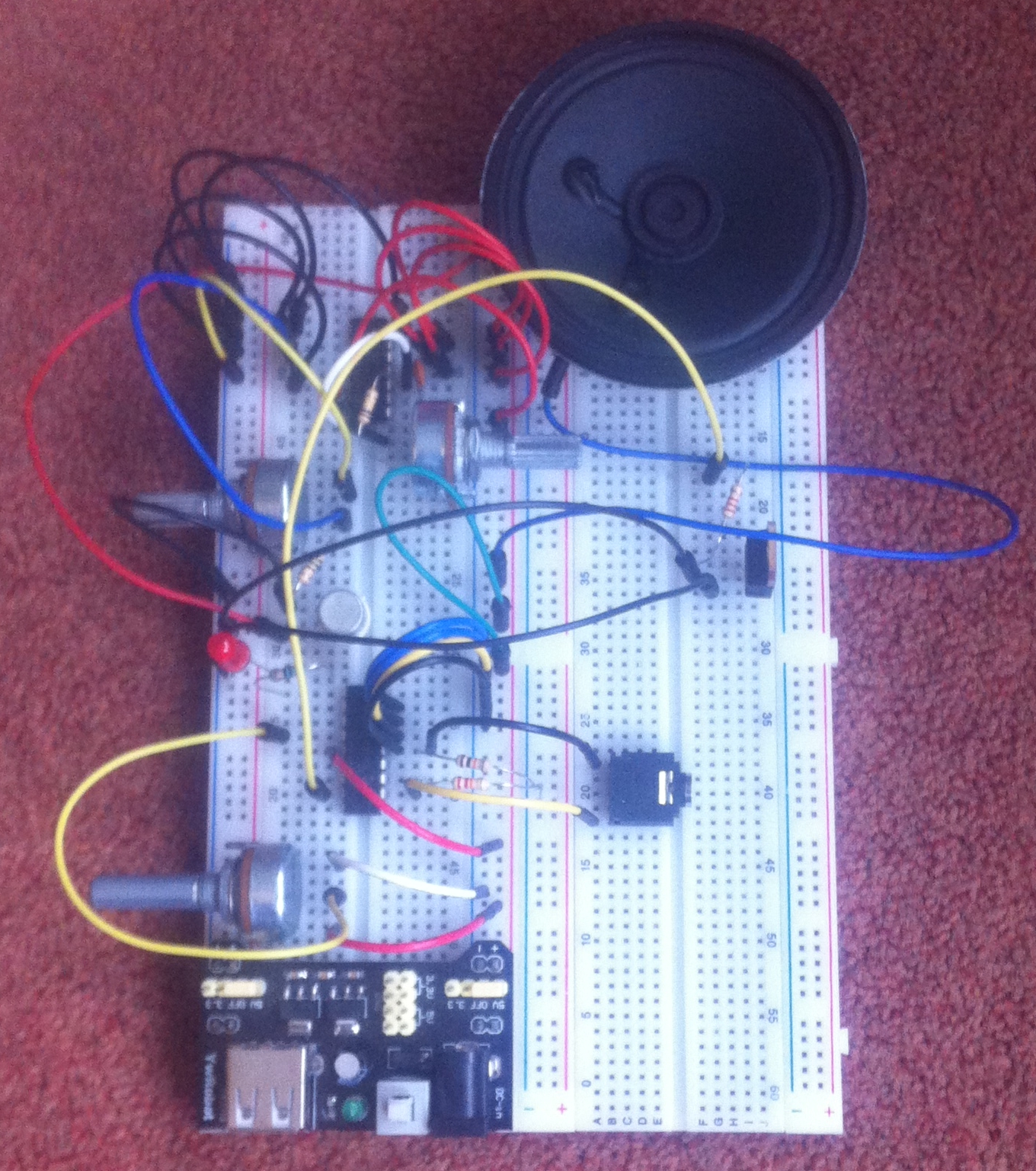
|  |  |
| --- | --- |
| **Pin** | **Function** |
| 1 – C.2 |  |
| 2 – C.3 | Serial Out – See programming circuit below. |
| 3 – C.4 | Serial In – See programming circuit below. |
| 4 – C.5 |  |
| 5 | 0V |
| 6 - B.0 | Letter switch – All switches are there to pull the associated pins low – to ground. The PICAXE had internal pull up resistors and they are enabled in software. |
| 7 – B.1 | Digit switch – See above. |
| 8 – B.2 | Punctuation switch – See above. |
| 9 – B.3 |  |
| 10 – B.4 | LED control – the pin goes to a 330 resistor, to the LED and the to ground. |
| 11 – B.5 |  |
| 12 – B.6 |  |
| 13 – B.7 | Speed control – the pin goes to the centre of a 10K potentiometer. The other two connections of the potentiometer are to +ve and ground respectively. |
| 14 | +ve |
| 15 – C.6 |  |
| 16 – C.7 |  |
| 17 – C.0 | Oscillator control – the pin goes to a 2K2 resistor and then to the base of the TIP120. |
| 18 – C.1 |  |

Programming circuit:



# Vero board Layout

The project started life on bread board that looked like:

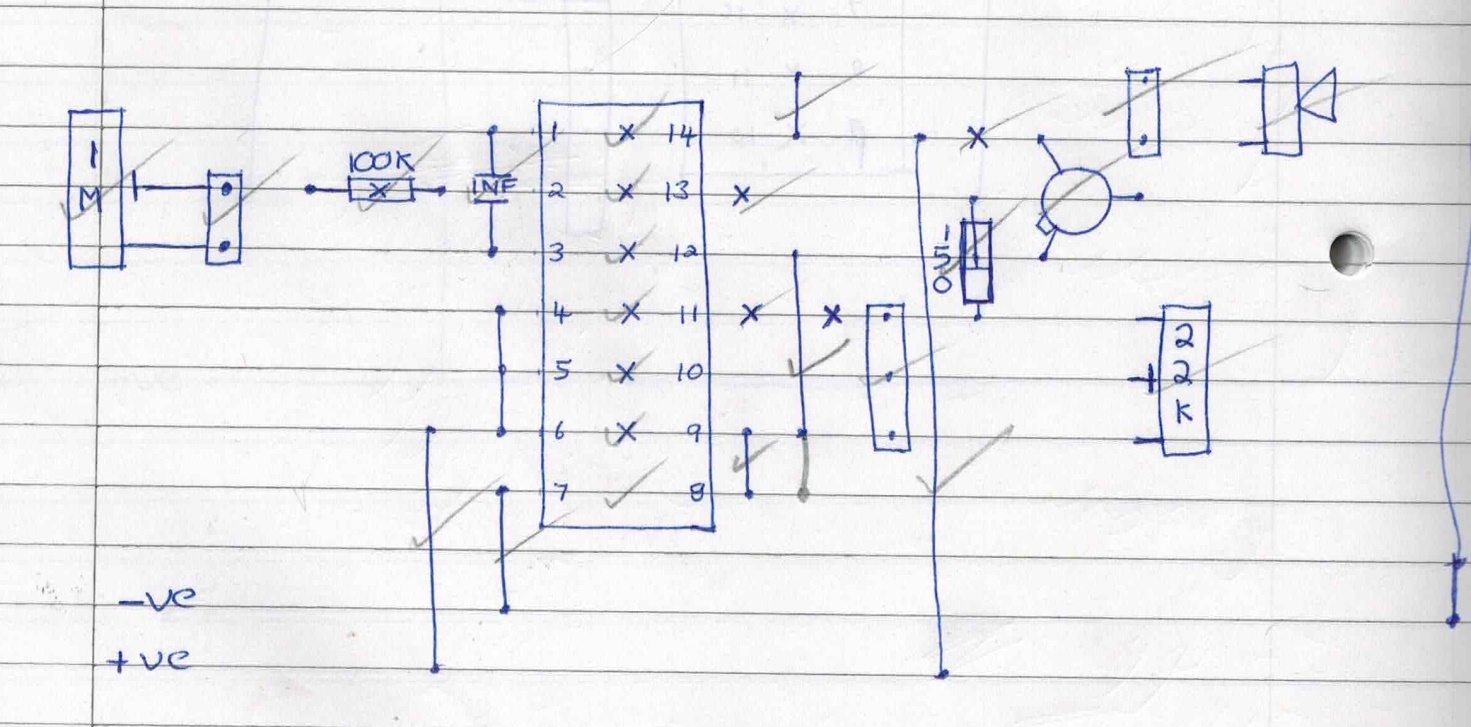


This all looks a bit slap dash but that’s the nature of bread board.

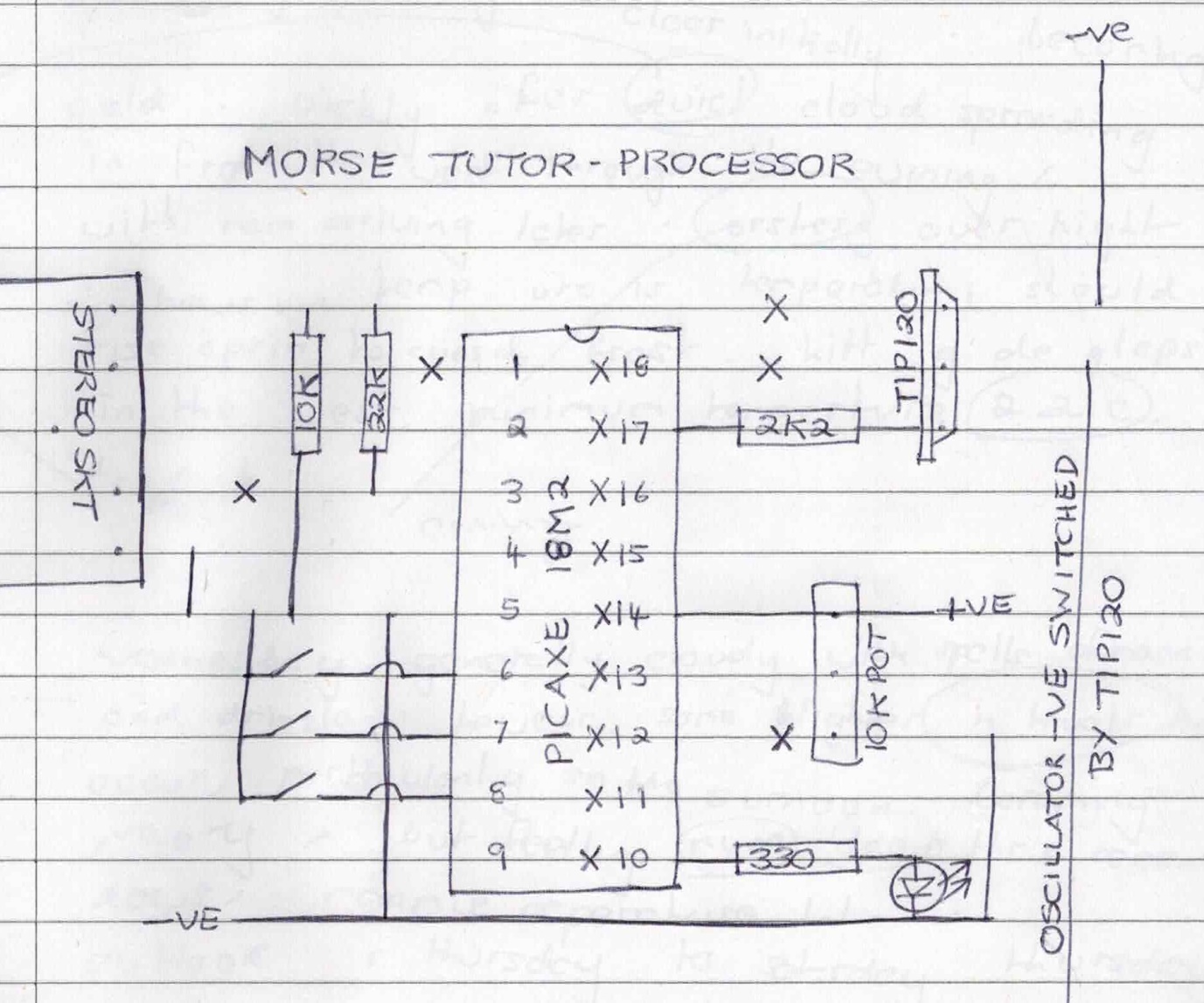
The Vero board layout was not very inspired and I’m sure that the current layout could be optimised for size. I basically split the Vero board into three distinct sections, the power supply, the processor, and finally the oscillator and placed them top down.

I started with the power supply as a simple standalone unit that could easily be tested. I stuck a diode in there to protect the circuit from reverse polarity too.

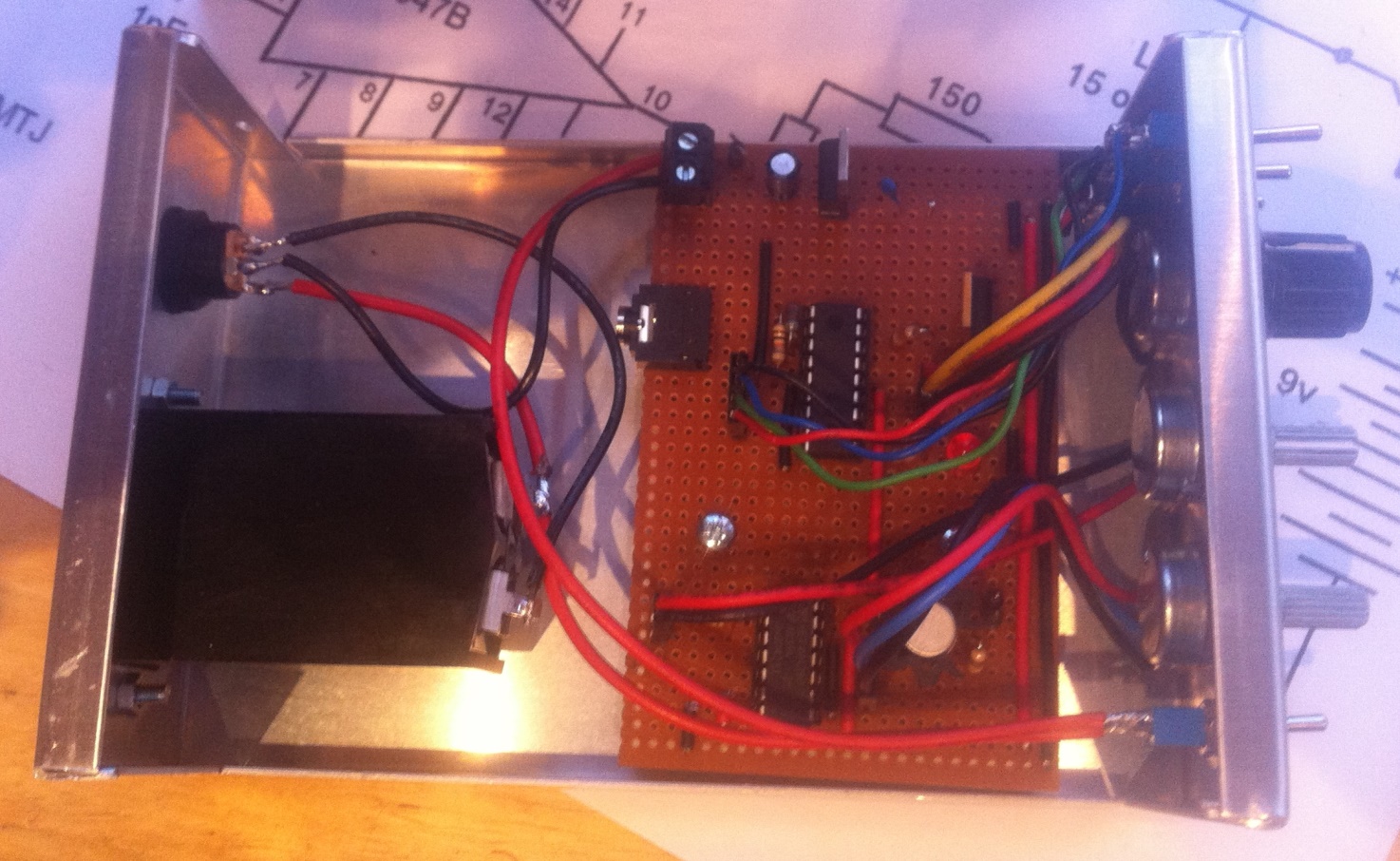
I next built the oscillator at the bottom of the board. Again it is a fairly standalone unit that that easily be tested. I used connectors to connect the potentiometers and speaker and it allowed for easy removal when adding the processor. Ignore the ticks in the image below, that’s just me ticking them off as I have done them.



Finally I put the processor in the middle. The programming socket could be left out but that would make further software updates difficult so it’s worth adding. The odd thing about the processor unit is that the Darlington pair is used to switch the –ve feed for the oscillator and not the +ve feed! I consider this a bit of a an odd approach but it works.



As for the final result, a picture paints a thousand words:



# List of parts

## Regulated Power Supply

|  |  |
| --- | --- |
| **What** | **Quantity** |
|  |  |
|  |  |
|  |  |

## Oscillator

|  |  |
| --- | --- |
| **What** | **Quantity** |
| IC 4048B | 1 |
| Matching IC socket | 1 |
| Potentiometer 1M Linear | 1 |
| Potentiometer 22K Linear | 1 |
| Resistor 100K | 1 |
| Resistor 150 | 1 |
| Transistor BFY51 | 1 |
| Capacitor 1nF | 1 |
| Speaker 15 Ohms (I used 8 Ohms) | 1 |

## Processor Unit

|  |  |
| --- | --- |
| **What** | **Quantity** |
| PICAXE 18M2 – Programmed | 1 |
| Matching IC Socket | 1 |
| Potentiometer 10K Linear | 1 |
| TIP120 Darlington Pair | 1 |
| LED | 1 |
| Resistor 330 | 1 |
| Resistor 2K2 | 1 |
| Resistor 10K | 1 |
| Resistor 22K | 1 |
| Switches | 3 |
| 3.5mm stereo socket PCB | 1 |

## Miscellaneous

|  |  |
| --- | --- |
| **What** | **Quantity** |
| Vero board (36 strips \* 24 holes) | 1 |
| Battery holder 9V | 1 |
| External power supply socket / plug | 1 |
| Power switch | 1 |
| Box | 1 |

# Possible enhancements

Below is a list of possible enhancements that might be considered in future versions.

1. An ear phone socket.
2. A Morse key socket. This would allow the device to be used as a simple Morse oscillator.

# References

<http://www.picaxe.com/> - this is the home of the Picaxe.