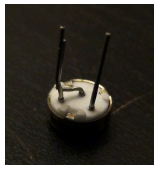
MicroModem Kit Quickstart Guide

This guide provides a few pointers on getting started with MicroModem, and how to connect the board and flash the firmware. The serial connection to the modem is 9600 baud, 8N1, no flow control.

Notes on the circuit

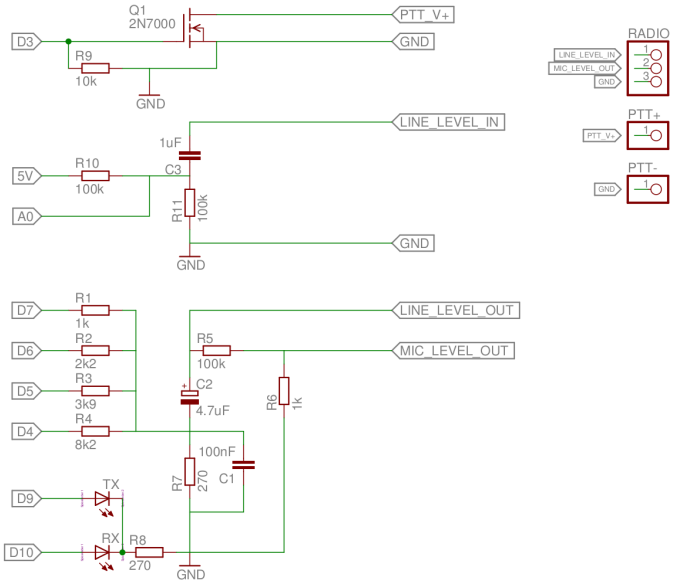
The circuit for MicroModem is really simple, and only needs a few common components. Here's a few notes and explanations about the circuit.



- Modulated output audio from the modem can be set from *microphone level* to *line level* voltages by manipulating the R5 trimpot. Connect two of the legs of the trimpot together as shown here, to make it work as a variable resistor.

- The PTT triggering part of the circuit is designed with the assumption of a Kenwood HT style PTT switch, namely: The ground pin on the **microphone** jack of the radio supplies about 3v, which will trigger PTT if it is connected to the ground pin of the **speaker** jack. The circuit accomplishes this by using a N-channel MOSFET to operate as a switch between the +3v and GND for PTT.

Have a look at the schematic here. The circuit is divided into three logical parts.



- The topmost part is the PTT triggering circuit.

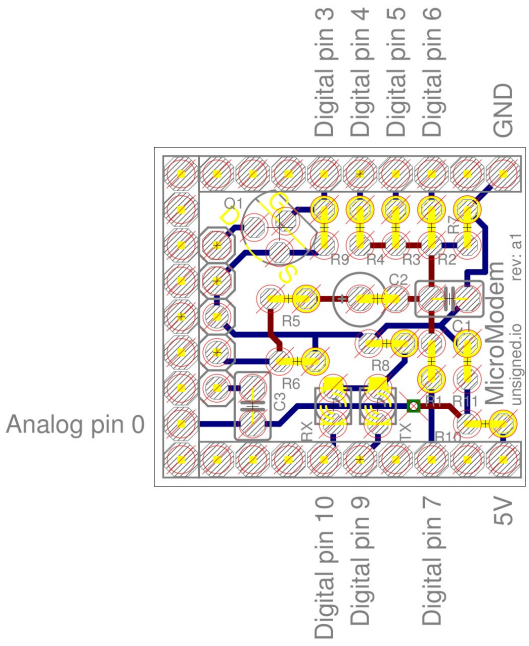
- The middle part is the input stage. This part AC-couples the input from the speaker jack of the radio and provides a 2.5v DC-bias, so we will be able to sample the audio with the ATmega's ADC. **R10** and **R11** constitutes a voltage divider that provides a 2.5v bias source from the ATmega's regulated 5V output.

- The bottom part is the output stage. This part is basically a very simple "soundcard". **R1** through **R4** constitutes a 4-bit string-resistor DAC. **C2** AC-couples the output, and **R6** and the **R5** trimpot forms a variable voltage divider, that supplies output levels from around 20mV to 1.5v peak-to-peak. Most radios wont be able to accept the line-level output generated directly by the ATMega, without distorting it to a point where it will be impossible to correctly demodulate again, in which case you will need to set the output to microphone level.

Partlist

To assemble the modem, place the parts into the designated positions on the board and solder! :)

Part	Value
C1	100nF
C2	4.7uF
C3	1uF
Q1	2N7000
R1	1k
R2	2k2
R3	3k9
R4	8k2
R5	100k trimpot
R6	1k
R7	270
R8	270
R9	10k
R10	100k
R11	100k
RX	LED BLUE
TX	LED ORANGE



Connecting to an Arduino

Here's an example of how to connect the assembled MicroModem board to an external MCU board. I will be using the Arduino UNO as an example, which should make it easy to adapt to other variants or even a breadboard ATmega328p. If you have a Microduino, the modem board fits right into the pin layout of that, so no need for connecting with wires. For connceting to an Arduino UNO board, just use the diagram above to connect the pads on the modem board to the corresponding Arduino pin.

Flashing a firmware

When you have built and connected the modem board, you will need to flash a MicroModem or MicroAPRS firmware to your MCU.

You can download firmwares at http://unsigned.io/projects/microaprs, or you can compile it yourself from the source code. I won't go into details about compiling here, but you can find information on that in the GitHub repository.

When you have a firmware (.hex) file, you can use `avrdude` to flash it. Here's an example of how to use it:

```
avrdude -p m328p -c arduino -P /dev/tty$1 -b 115200 -F -U flash:w:images/Modem.hex
```

The firmware should the be flashed and ready to use! There's plenty of other programs for loading .hex files onto your MCU, so if you use another one, that's also just fine.