# MicroModem User Manual

Thank you very much for buying this product! If you have any questions, suggestions, criticisms or good ideas, I'd love to hear from you! This guide provides a few pointers on getting started with MicroModem, and how to connect the board and flash the firmware.

## Connecting host equipment to the modem

The serial connection to the modem is 9600 baud, 8N1, no flow control. You should be able to set these parameters in whatever program you will connect the modem with, or from the hardware you are connecting to the modem. If you need to use another baud-rate, it is possible to change this quite easily in the source-code and recompile the firmware. The modem has been tested with rates up to 115200 baud. In KISS mode, various parameters can be configured by the host program over the USB connection, but this comes with a few caveats, so if you intend to do so, please see: http://unsigned.io/using-kiss-settings

## Connecting to a radio

To connect the modem to a radio, you must connect the **Audio Out** port on the modem to the microphone input of your radio, and the **Audio In** port on the modem, to the speaker/headphones output of your radio. Also connect the **GND** port on the modem to the speaker or mic (depends on radio) ground. With a small flat-headed screwdriver, trim the potentiometer on the board to adjust the output level to an acceptable level for your radio. Turned all the way counter-clockwise, the output level will be around 1.5V peak-to-peak. Turned all the way clockwise, the output level will be around 15mV peak-to-peak. This roughly equates to a range going from line-level to microphone-level. Optionally, connect the PTT ports according to the PTT mode of your radio. If your radio uses Kenwood-style PTT triggering, supplying a positive voltage on the ground pin of the microphone plug, connect that to the PTT_V+ pin. If your radio accepts a logic-level PTT signal, connect the **PTT** port of the modem to the PTT input of your radio.
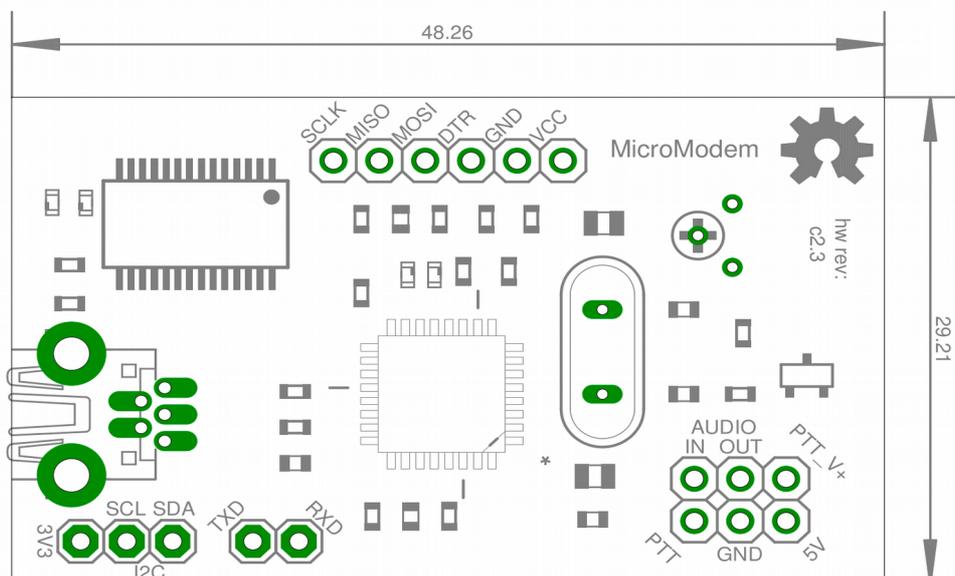
# Expansion ports

In addition to the radio connector ports just discussed, the modem supplies several extra expansion ports. Please refer to the previous illustration for the location of these ports. The UART serial port is available on the **TXD** and **RXD** ports. An **I2C** port is also available, marked **SCL** and **SDA**. Furthermore, an **SPI** port is avaible, marked **SCLK**, **MISO** and **MOSI**. An extra **GND** connector is made available, along with two 5V/VCC ports marked **VCC** and **5V**. There is also a 3.3V supply broken out, labeled **3V3**. The max supply for this port is **50mA**.

These extra ports are not necessarily used by your chosen firmware, but can be used for future functionality, or you can use these yourself, if you plan to write your own firmware. All the pins listed are also available as general purpose digital and analog IO pins. The following list maps the port names to Arduino IDE pin designations:

- RXD       =     Digital pin  0
- TXD       =     Digital pin  1
- MOSI      =     Digital pin 11
- MISO      =     Digital pin 12
- SCLK      =     Digital pin 13
- SDA       =     Analog  pin  4
- SCL       =     Analog  pin  5

# Operating environment

The modem uses all high-quality components, and should last for many decades with normal use. If you plan to use the modem outdoors, I recommend keeping the modem within an operating temperature of **-20°C** to **70°C**. The modem should also be kept free of condensing conditions, so that the bare PCB is not exposed to moisture. The same goes for rain or other sources of moisture. If you will be operating the modem in a wet environment, I recommend placing it in an IP55 rated housing. I you need to operate the modem totally submerged, applying a full epoxy-coating is recommended.

# Flashing a firmware

The modem comes preflashed with the newest version of the firmware you choose when ordering. If you want to use another firmware on the modem, just flash it from your computer. You can do this as many times as you need, and it is not a difficult or risky process. You can download firmwares at [http://unsigned.io/projects/micromodem](http://unsigned.io/projects/micromodem), 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 repositories. 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/ttyUSB0 -b 115200 -U
flash:w:YourFirmwareFile.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. A few programs with a nice GUI also exist, if you don't feel comfortable using the command line.

## Making your own firmware

You can use the the source code for any of the available firmwares as a basis for your own. If you prefer to use the Arduino IDE, you can use LibAPRS to easily make APRS-oriented firmwares straight from the Arduino IDE. LibAPRS is available at http://unsigned.io/projects/libaprs/. If you have any questions or problems, try asking in the forums!

## Staying up to date

All MicroModem firmwares and libraries are in active development, and are updated regularly. Check http://unsigned.io for updates and news!

## Getting help

If you need help for anything, a great place to start is the forums at http://unsigned.io/forum! You are also very welcome to email me at mark@unsigned.io, and I will get back to you ASAP!

## Technical Specifications

Here's a list of technical specifications, included for your reference:

- ATmega328p @ 16MHz (2KB RAM / 32KB flash)
- Powered from USB or external regulated 5V source
- 51mA idle power consumption
- Operating range: -20°C to 70°C (non-condensing)
- Modulated analog output level adjustable from approximately 15mV to 1.5V peak-to-peak
- Minimum analog input level for good decode is about 100mV peak-to-peak
- ADC running on 3.3v reference
- Arduino compatible (You can program the board from the Arduino IDE, over USB)


**Thanks a lot for buying this product!**
I am very grateful for your business, and I sincerely hope my product will live up to your expectations.


I have designed and crafted this item with all my love and care, and by purchasing from me, you are directly supporting independent open source hardware design.
Thank you so much!


- Mark

ANALOG_OUT

R6 1k
C2 4.7uF
C1 100nF
R7 270
R5 100K
GND
PTT_V+
Q1 2N7000
GND
R9 10k
PTT_PIN

R4 8k2
R3 3k9
R2 2k2
R1 1k

SDA
SCL
RESET
MCU_RX
MCU_TX
PTT_PIN

3V3
SDA
SCL
MCU_RX
MCU_TX
3V3 SDA SCL RXD TXD

ICSP
5V GND RESET
PRG
MOSI MISO SCK

ATMEGA328P-AU
328P

23 PC0
24 PC1
25 PC2
26 PC3
27 PC4
28 PC5
29 PC6
30 PD0
31 PD1
32 PD2
1 PD3
2 PD4
9 PD5
10 PD6
11 PD7

VCC_2 4
VCC 6
AVCC 18
ADC6 19
ADC7 22
AREF 20
PB0 12
PB1 13
PB2 14
PB3 15
PB4 16
PB5 17
PB6 7
PB7 8
GND_2 3
GND_3 5
GND 21

5V
3V3
GND

MOSI MISO SCK

XTAL 16MHZ
C5 22pF
C4 22pF
GND

TX
RX

ANALOG_IN
C3 1uF

R8 270
R16 1k
3V3
R11 100K
R10 100K
GND

5V
C8 10uF
GND
C7
C6 100nF
R14 10k
RESET
GND

JP1
1 2 3
JP2
1 2 3
ANALOG_IN
ANALOG_OUT
PTT_V+
5V
GND
PTT_PIN

IC1 FT232RL
MCU_RX
MCU_TX
R12 10
RESET
C6 100nF
R15 1k
R13 270
URX
UTX
5V

TXD 1
RXD 5
RTS 3
CTS 11
DTR 2
DSR 9
DCD 10
RI 6
CBUS0 23
CBUS1 22
CBUS2 13
CBUS3 14
CBUS4 12
TEST 26
GND 7
GND 18
GND 21

VCC 20
VCCIO 4
RESET 19
OSCI 27
OSCO 28
3V3OUT 17
USBDP 15
USBDM 16
GND 25

GND
3V3
GND

USB
VBUS
D+
D-
ID
GND
5V
GND