Beginner's Guide to RTTY Operation

SEMDXA Monthly Meeting
May 8, 2015

Larry Gauthier, K8UT

Beginner's Guide to RTTY

- Agenda
  - Digital Mode Anatomy

    Compare CW to RTTY
  - Preparing to Operate RTTY

    Create Your Shopping List
  - Connecting the RTTY Components

    Insert Tab A in Slot B
  - Operating RTTY Using MMTTY

    Can You Hear Me Now?
  - Stuff to Avoid

    Learn from My Mistakes

CW: The Morse Code – 1836

- Morse Code Character Set (42)
  - Upper-case characters: A – Z
  - Numbers: 0 – 9
  - Some punctuation:

    / , . ? – (plus newest @ for email)
  - Prosigns:
    - Error, Break, End-of-Transmission, End-of-QSO, Wait

CW: The Morse Code

- Morse Code Architecture
  - Asynchronous transmission
    - No prior agreement on speed
    - CW speed is deduced from element times
      - Dots = 1 timing element long
      - Dashes = 3 timing elements long
      - Intra-character spacing = 1 timing element long
      - Inter-character spacing = 3 timing elements long
      - Inter-word spacing = 6 timing elements long
  - Variable character lengths
    - Shortest character = E = 1 timing element long
    - Longest character = 0 = 19 timing elements long

CW: The Morse Code

- Speed in measured in Words Per Minute (WPM)
  - Special word : ***Paris*** (50 time elements)

    .–––. .– .–. .. ...

  - **At 60 WPM a dot = 20 milliseconds, a dash = 60 milliseconds**
    - Formula: `timing element length = 1.2/words per minute`
- Sample CW transmission at 60 WPM

  CQ DE N8SL (8 transmitted characters, sent 3 times)

–.–. ––.– –.. . –. –––.. ... .–..

C Q D E N 8 S L

RTTY: The Baudot Code -1870

- Baudot Code Character Set (54)
  - Upper-case characters: A – Z
  - Numbers: 0 – 9
  - More punctuation:

    / , . ? – & # ! ‘ “ ( ) : ; *<space>* *<cr>* *<lf>* *<bell>*
  - Control Characters:
    - Shift-to-Letters, Shift-to-Figures (more on these later)

RTTY: The Baudot Code

- Baudot Code Architecture
  - Pseudo-synchronous transmission
    - Requires prior agreement on speed
      - All characters are 5 bits long
      - All characters are preceded by a start bit (usually 1x)
      - All characters are succeeded by a stop bit (usually 1.5x)
      - Each character is precisely 7.5 bits long
  - RTTY protocol has no error correction
    - Many sources of reception errors
      - QRM, QRN, fading, multi-path, tropospheric distortion

RTTY: The Baudot Code

- Baudot Character Set (54) ????
- How do you get 54 characters from 5 bits?
  - Maximum from 5 bits would be 31 characters
- Use Shift-Letters and Shift-Figures commands
  - Letters = A – Z and <space> <cr> <lf>
  - Figures = 0 – 9 and punctuation plus <space> <cr> <lf> <bell>

Strange expressions appear on your screen when the shift commands get garbled in transmission

RTTY: The Baudot Code

- RTTY = Frequency Shift with a Constant Carrier
  - Unlike CW with its intermittent carrier
  - Frequency shift on Ham Bands = 170 Hz
    - Defaults are Mark Hz = 2125, Space Hz = 2295
  - Blank time between characters is filled with "*diddles*"
    - Non-printing characters that maintain synchronization
    - Results in 100% duty cycle during transmit

RTTY: The Baudot Code

- RTTY Speed is Measured in Baud
  - Four FCC approved amateur baud rates:
    - 45.45 (60wpm), 50 (67wpm), 56 (75 wpm), 75 (100 wpm)
    - Most amateur RTTY operation is 45.45 baud (60 wpm)
  - At 45.45 baud, bit length is 22.002 milliseconds
  - Formula: `bit length in milliseconds = 1 / baud rate`
- Sample RTTY transmission at 45.45 baud

  CQ DE N8SL (12 transmitted characters, sent 3 times)

```
-01110=-10111=-00100=-01001=-00001=-00100=
C          Q        space    D         E        space
-01100=-11011=-00110=-11111=-00101=-10010=
 N         FIGS     8        LTRS     S        L
```

Preparing to Operate RTTY

- What do I need to buy?
- Popular scenarios:
  - RTTY-equipped transceiver
  - PC with RTTY software and a transceiver
  - PC with RTTY software, a TNC, and a transceiver

Preparing to Operate RTTY

- What do I need to buy?
  - RTTY-equipped transceiver
    - All you need is the radio, and maybe a keyboard
  - PC with RTTY software and a transceiver
    - RTTY software (most programs are free)
    - Cables between PC and the radio
      - Homebrew, commercial cables, commercial interfaces
  - PC with RTTY software, a TNC, and a transceiver
    - RTTY Software (must have drivers for your TNC)
    - A TNC (Terminal Node Controller) ***or equivalent****
    - Cables from PC-to-TNC, TNC-to-Radio

Connecting the Components

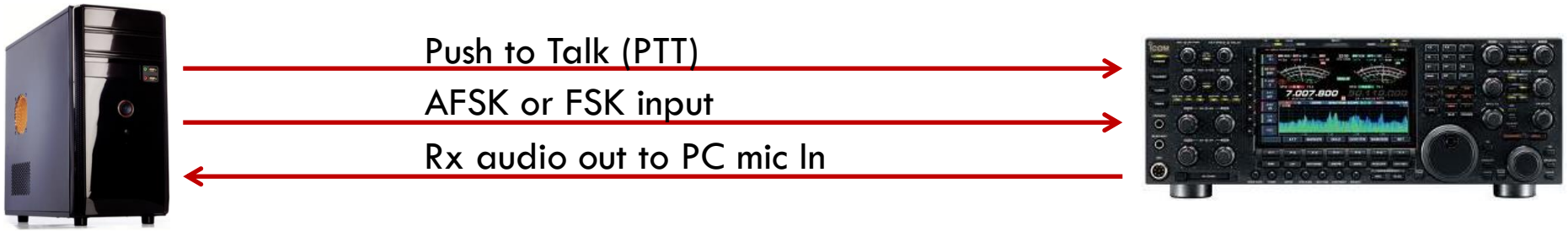
- RTTY-equipped transceiver
  - RTTY encoder and decoder are built-in
  - Can be used with an optional keyboard
  - Disadvantages
    - Expensive radio
    - Minimal functionality (*)
    - No log integration
    - Limited function keys
    - Poor contest exchange

This is the least popular method (easy, expensive, few features)

Connecting the Components

- PC with RTTY software and a transceiver
  - Software: MMTTY, fldigi, WinWarbler… others
  - Homebrew or commercial interface and cables

This is the most popular method
(easy, cheap, full-featured)



Connecting the Components

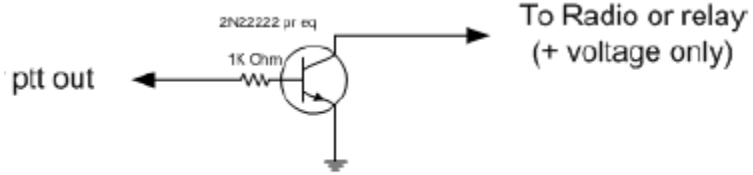
- PC with RTTY software and a transceiver
- Options for connecting receive audio
  - A microphone plugged into PC mic input
  - Direct front/rear panel jack to PC mic/aux input
  - Direct but add an audio isolation transformer
  - USB cable with audio codecs
  - Commercial interfaces
    - microHAM, West Mountain Radio, MFJ, others…

Rx audio out to PC mic In

Connecting the Components

- PC with RTTY software and a transceiver
- Options for connecting **PTT for transmit**
  - PC serial port DTR/RTS pins to ACC jack Tx line
    - Requires simple one resistor, one transistor buffer
  - PC USB-to-serial adapter (FTDI chipset recommended)
    - Requires simple one resistor, one transistor buffer
  - Commercial interface
  - **VOX* – if using AFSK. No PTT circuit needed*

Push to Talk (PTT)



Serial port cw or ptt interface

Connecting the Components

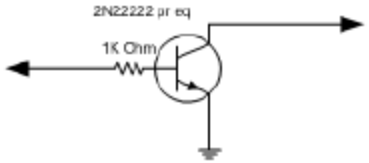
- PC with RTTY software and a transceiver
- Options for connecting **AFSK transmit**
  - Direct front panel mic input or rear panel aux jack to PC speaker or Aux audio output
  - Direct but add an audio isolation transformer
  - USB cable with audio codecs (radios with USB out)
  - Commercial interfaces

AFSK input

Connecting the Components

- PC with RTTY software and a transceiver
- Options for connecting **FSK transmit**
  - PC serial port Data out pin to ACC jack FSK line
    - Requires simple one resistor, one transistor buffer
  - PC USB-to-serial adapter (FTDI chipset recommended)
  - Commercial interface

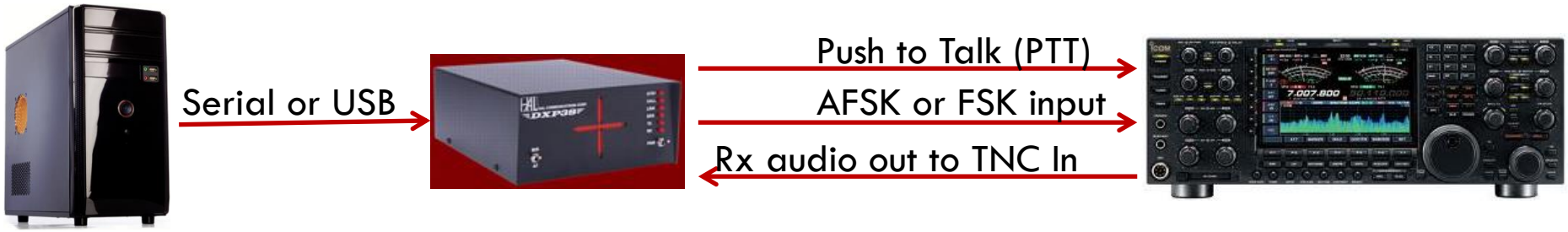
FSK input



Connecting the Components

- PC with RTTY software a TNC and a transceiver
- Options for connecting a TNC
  - Software: MMTTY, fldigi, WinWarbler… others
    - Requires driver support for your model TNC
  - Serial cable from PC to TNC (serial port or USB)
  - Three cables between TNC and transceiver

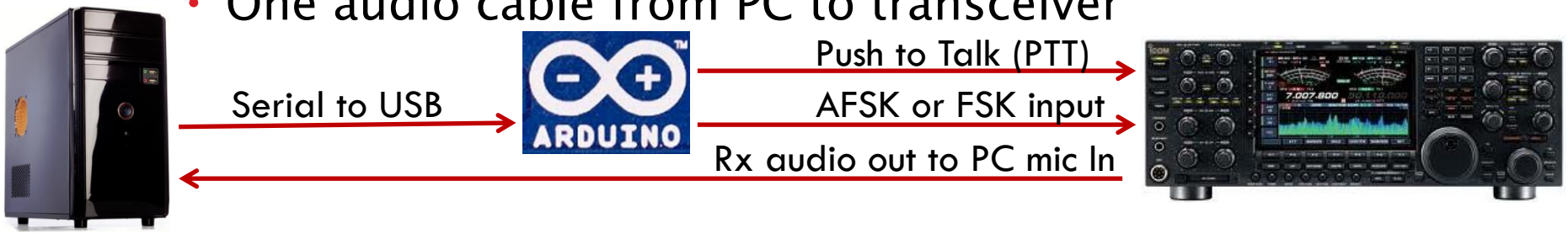
    (see previous slide)



The middle of the popularity ratings
(hard, expensive, full-featured)

Connecting the Components

- PC with RTTY software a TNC and a transceiver
- Options for connecting *"or Equivalent"*
  - TNC is replaced by Single Board Computer: Arduino
  - Software: On PC, MMTTY on N1MM and 2Tone
  - Software: On Arduino, tinyFSK (free) by Andy K0SM
    - Serial cable from PC to SBC (serial port or USB)
    - Two cables between TNC and transceiver
    - One audio cable from PC to transceiver



2Tone interface now available!

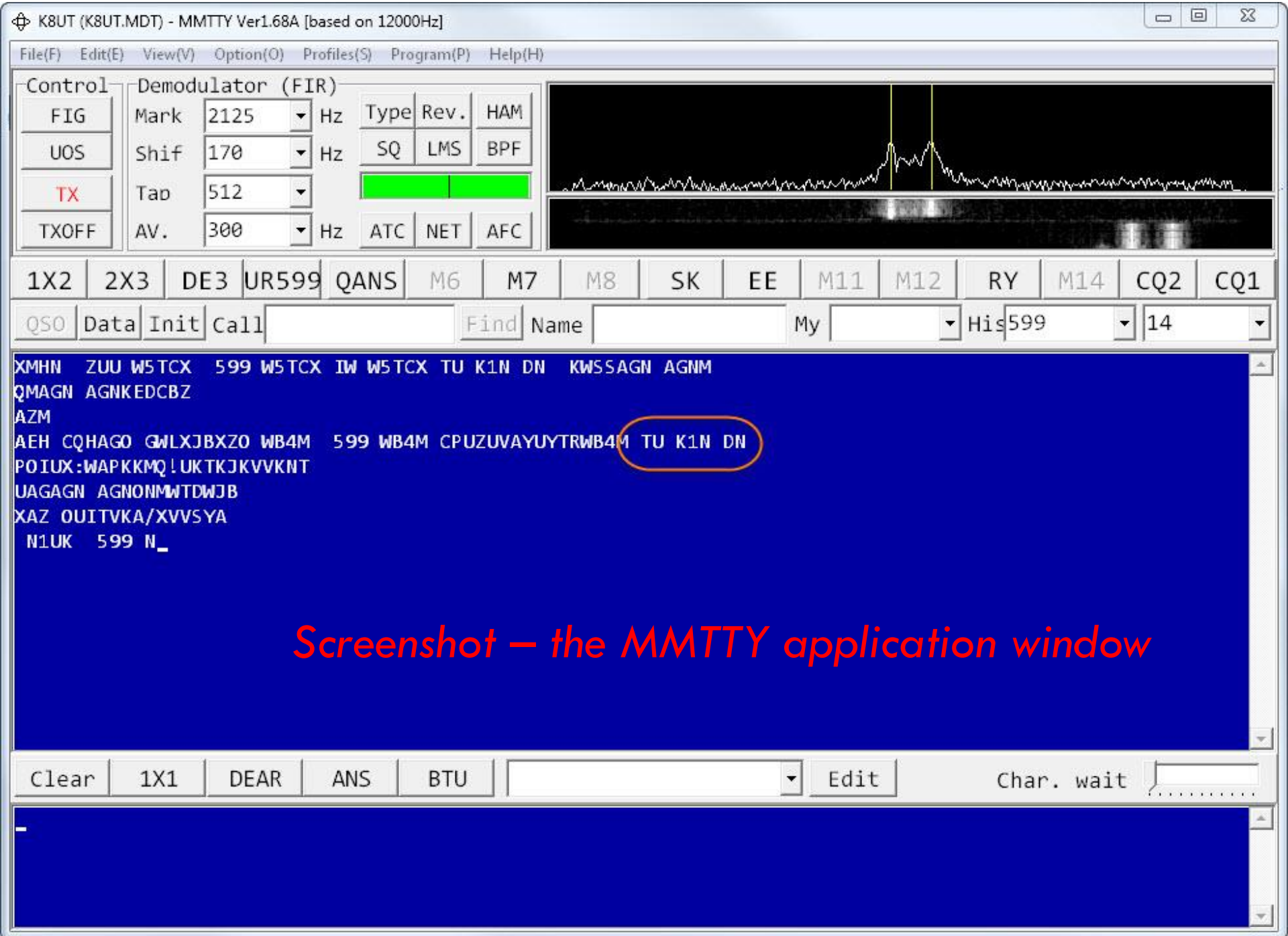
Raspberry Pi version coming soon!

Operating RTTY with MMTTY

- Operating RTTY Using MMTTY

  Can You Hear Me Now?

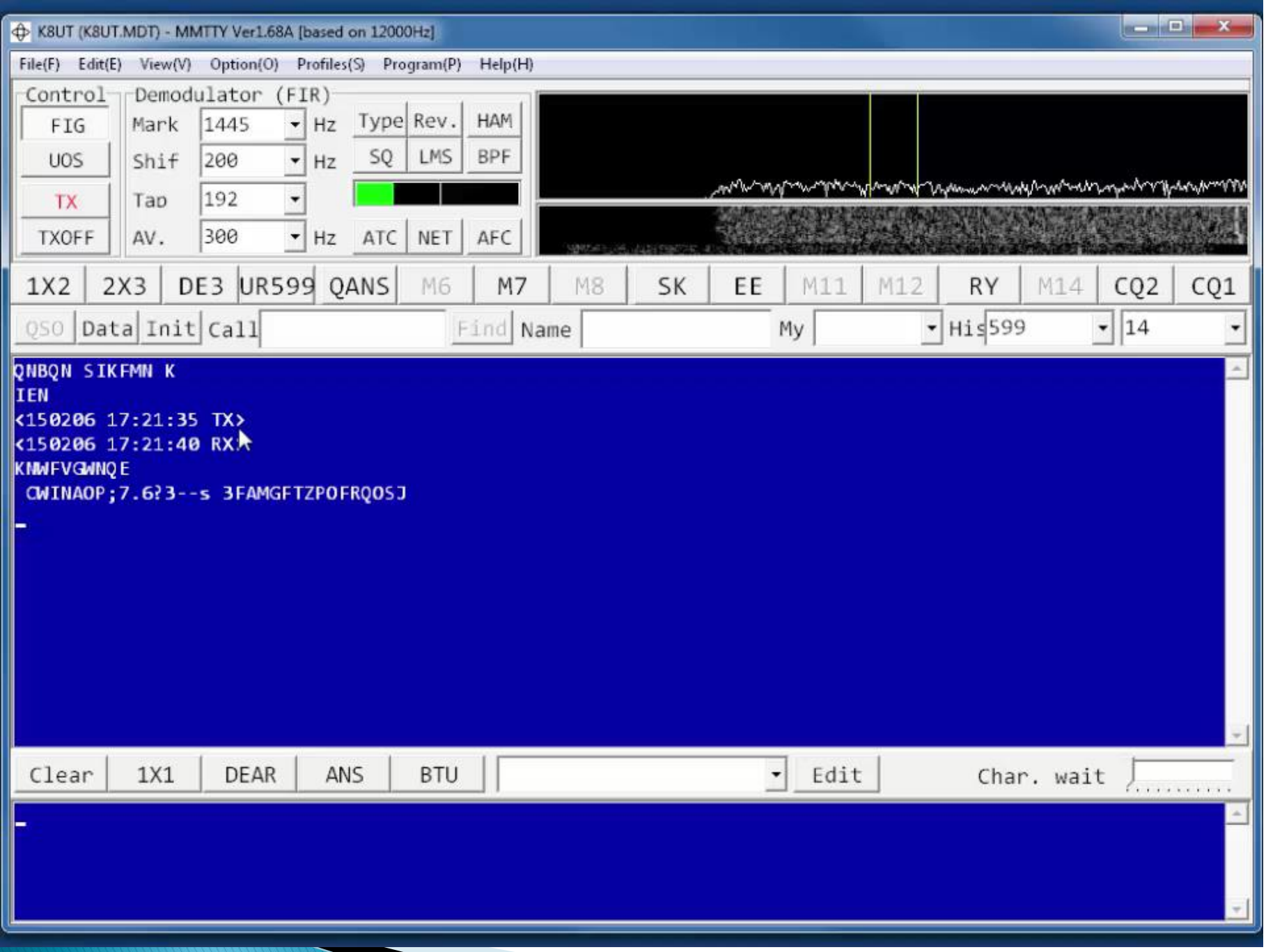
  - MMTTY as a stand-alone program
    - But MMTTY can be used as a slave app beneath many logging programs (N1MM, WinTest, DXLabs, HRD, WinWarbler, DX4Win, …)



Screenshot – the MMTTY application window

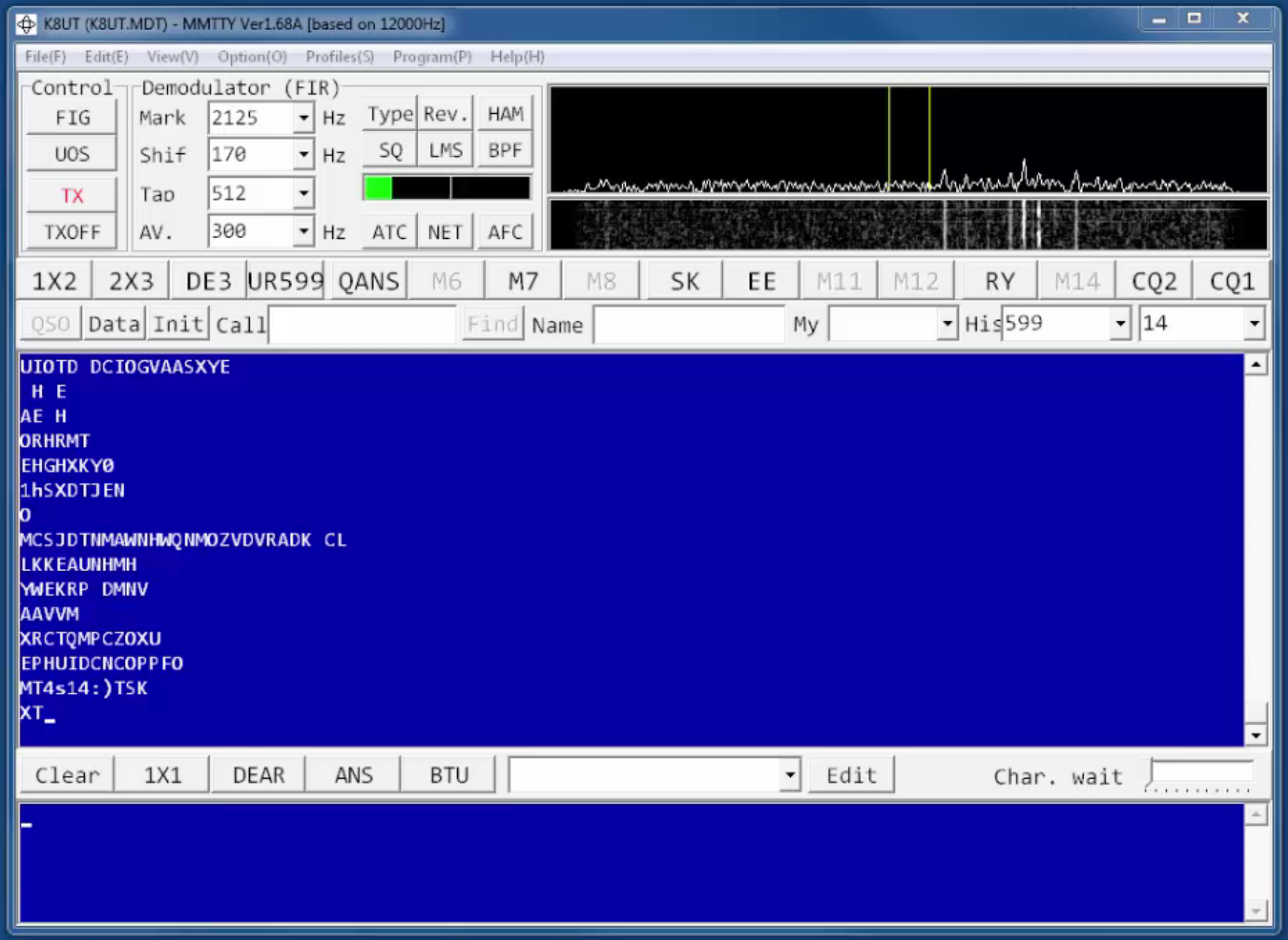
Operating RTTY with MMTTY

- Step 1 – MMTTY Go / No-Go Test
  - Is receive working?
  - Is transmit working?


K8UT (K8UT.MDT) - MMTTY Ver1.68A [based on 12000Hz]
File(F) Edit(E) View(V) Option(O) Profiles(S) Program(P) Help(H)
Control
FIG
UOS
TX
TXOFF
Demodulator (FIR)
Mark 1445 Hz
Shif 200 Hz
Tap 192
AV. 300 Hz
Type Rev. HAM
SQ LMS BPF
ATC NET AFC
1X2 2X3 DE3 UR599 QANS M6 M7 M8 SK EE M11 M12 RY M14 CQ2 CQ1
QSO Data Init Call Find Name My His 599 14
QNBQN SIKFMN K
IEN
<150206 17:21:35 TX>
<150206 17:21:40 RX>
KNWFVGWNQE
CWINAOP;7.6?3--s 3FAMGFTZPOFRQOSJ
Clear 1X1 DEAR ANS BTU Edit Char. wait

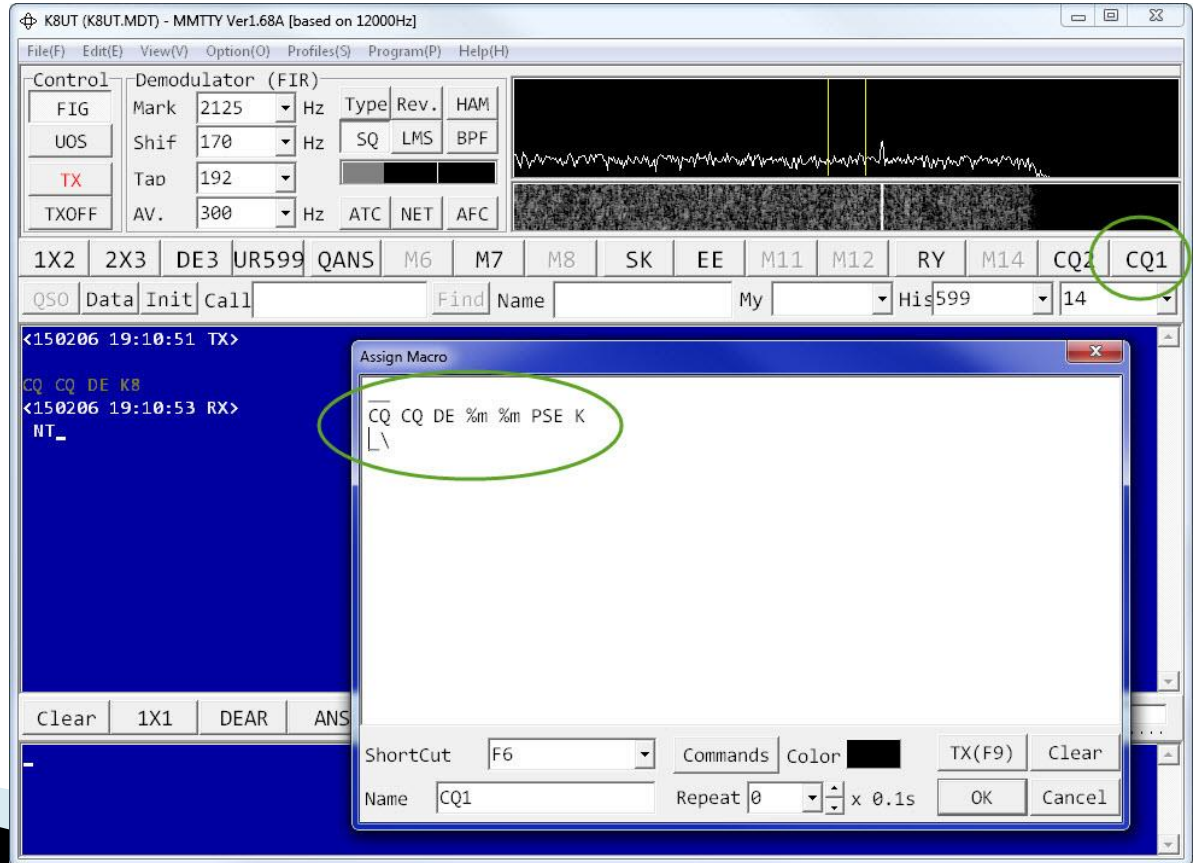
Operating RTTY with MMTTY

- Step 2 – Tune and Decode a RTTY Signal


K8UT (K8UT.MDT) - MMTTY Ver1.68A [based on 12000Hz]
File(F) Edit(E) View(V) Option(O) Profiles(S) Program(P) Help(H)
Control
FIG
UOS
TX
TXOFF
Demodulator (FIR)
Mark 2125 Hz
Shif 170 Hz
Tap 512
AV. 300 Hz
Type Rev. HAM
SQ LMS BPF
ATC NET AFC
1X2 2X3 DE3 UR599 QANS M6 M7 M8 SK EE M11 M12 RY M14 CQ2 CQ1
QSO Data Init Call Find Name My His 599 14
UIOTD DCIOGVAASXYE
H E
AE H
ORHRMT
EHGHXKY0
1hSXDTJEN
O
MCSJDTNMAWNHWQNMOZVDVRADK CL
LKKEAUNHMH
YWEKRP DMNV
AAVVM
XRCTQMPCZOXU
EPHUIDCNCOPPFO
MT4s14:)TSK
XT
Clear 1X1 DEAR ANS BTU Edit Char. wait

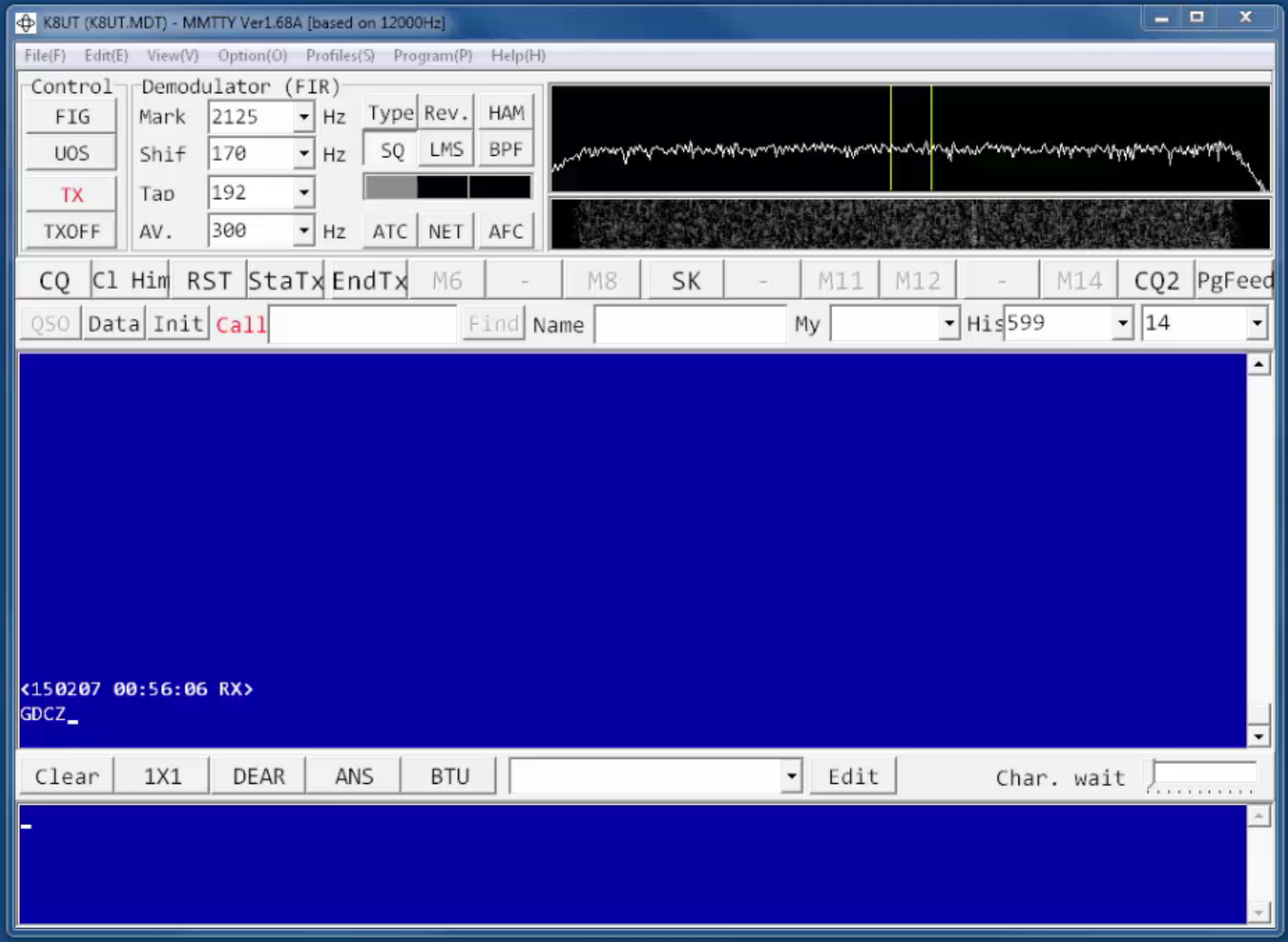
Operating RTTY with MMTTY

- Step 3 – My First QSO
  - BUT – before doing that…
    - Examine/adjust the Function Key assignments



Operating RTTY with MMTTY

- Step 3 – My First QSO
  - BUT – before doing that…
    - Examine/adjust the Function Key assignments
      - `Call CQ` `CQ CQ DE %m %m PSE K \`
      - `Call CQing station` `%c %c DE %m %m %m K \`
      - `Send signal report` `TU UR %r %r DE %m K \`
      - `Start Tx` `%c DE %m FB %n`
      - `End Tx` `BTU %n %c DE %m K \`
      - `End QSO` `73 %n %c DE %m SK \`


K8UT (K8UT.MDT) - MMTTY Ver1.68A [based on 12000Hz]
File(F) Edit(E) View(V) Option(O) Profiles(S) Program(P) Help(H)
Control
FIG
UOS
TX
TXOFF
Demodulator (FIR)
Mark 2125 Hz
Shif 170 Hz
Tap 192
AV. 300 Hz
Type Rev. HAM
SQ LMS BPF
ATC NET AFC
CQ Cl Him RST StaTx EndTx M6 - M8 SK - M11 M12 - M14 CQ2 PgFeed
QSO Data Init Call Find Name My His 599 14
<150207 00:56:06 RX>
GDCZ
Clear 1X1 DEAR ANS BTU Edit Char. wait

Operating RTTY with MMTTY

- Learn from My Mistakes
  - 26,551 RTTY QSOs
  - 330 DXCC Countries Worked/Confirmed

Stuff to Avoid

- Learn from My Mistakes
  - Do not try to decode non-ham RTTY signals
    - Non-standard baud and shift, usually encrypted
  - Do not operate out-of-band for US stations
    - Understand *FCC Band Allocations* **vs** *ARRL Band Plans*
      - Post an ARRL RTTY Band Plan map in your shack
      - Some WARC bands have narrow (10 KHz!) RTTY plans
    - Popular RTTY frequencies – just remember #84
      - 3.584, 7.034(DX)/7.084(US), 14.084, 21.084, 28.084
  - Examine the advantages and disadvantages of AFSK vs FSK operation
    - If you choose AFSK, monitor your ALC and audio levels

Stuff to Avoid

- Learn from My Mistakes
  - Using AFC is easy, but…
    - When using FSK results in RX / TX frequency skew
    - With AFSK, can shift frequencies down enough for 2nd harmonics to be in the Tx passband
  - Reverse RTTY – "You're Upside Down"
    - Some radios operate RTTY from USB, others from LSB
      - Mark/Space = Low/Hi *vs* Mark/Space = Hi/Low
  - Smoke test – power ratings for radio, amp, antenna
    - Unlike phone or CW, RTTY is continuous duty Tx

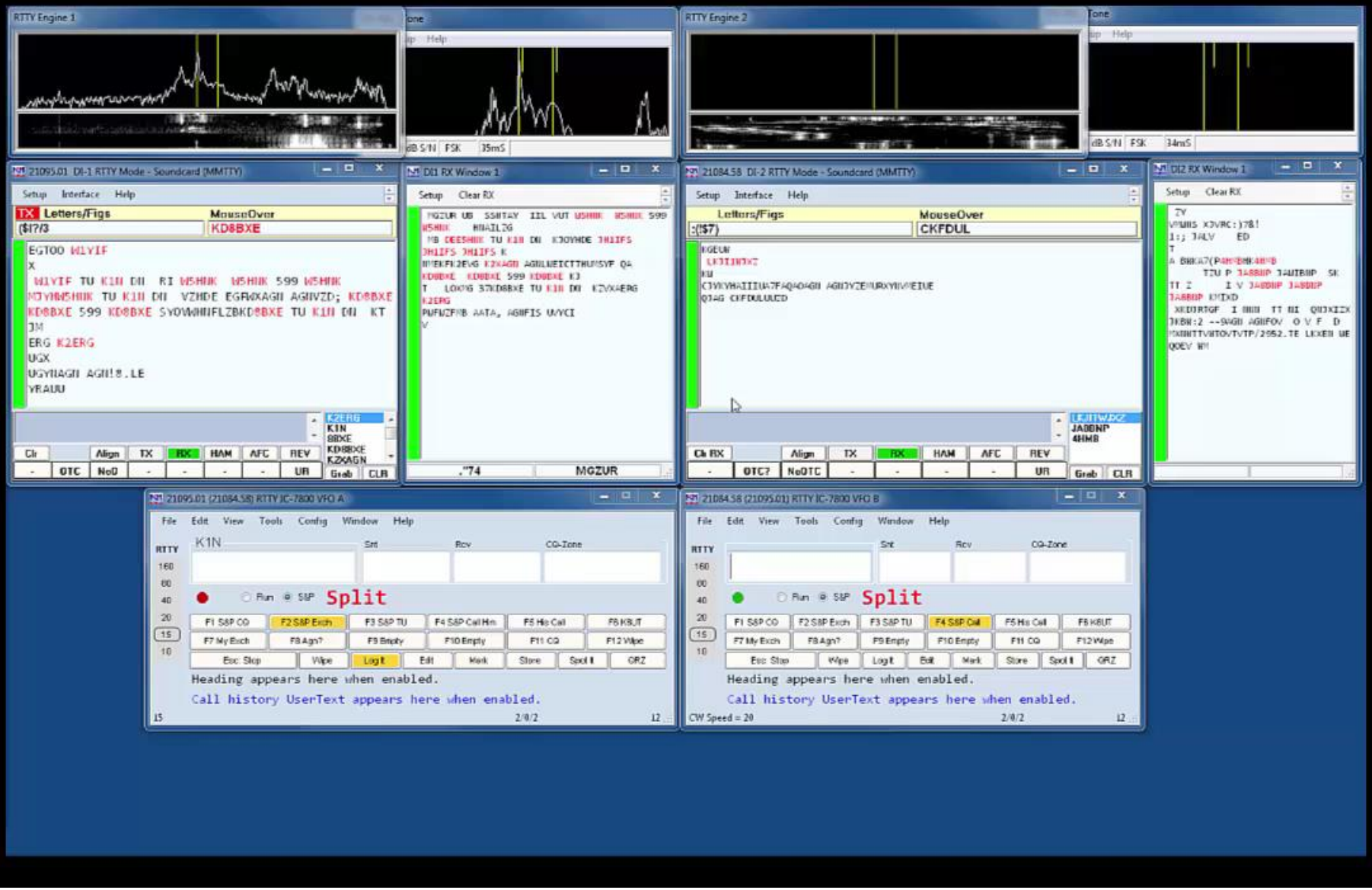
Stuff to Avoid

- Learn from My Mistakes
  - Avoid issues with USB-to-serial adapters
    - Generally FTDI chipsets are your safest choice
  - Recognize shift-LTRS and shift-FIGS problems
    - Most likely replacement: adjacent row of keys
    - When you see letters and expected numbers, or see numbers and expected letters
    - Most common examples
      - Signal report: **TOO TOO** (599 599)
      - Contest serial number: **PPQ PPQ** (001 001)
      - You own callsign: **K875 K875** (K8UT K8UT)

Beginner's Guide to RTTY Operation

The End … almost

Larry Gauthier, K8UT



K8UT's contest RTTY configuration for SO2V