NanoIO

An FSK / CW interface based on the Arduino nano. Sketch includes an FSK modulator, a CW computer interface, and a CW iambic-A/B keyer.

**FSK Specifications**

- 5 bit Baudot
- baud rates 45.45, 50, 75 and 100

**CW Specifications:**

- 5 to 100 WPM
- dash/dot ratio adjustable 2.5 to 3.5
- in-line increment decrement WPM using ^ and | characters
- incremental size user adjustable

PTT signal generated by Arduino

**Both**

- an internal buffer of 200 characters is available for buffered transmit.

**Hardware requirement**

- Arduino nano or compatible (author used nano from Elegoo)
- LTV-847 quad opto-isolator
- 4 620 ohm ¼ watt resistor (should work with 500 < R < 820 ohm)
- suitable connectors to interface to transceiver
Default pin assignments defined in Arduino Sketch for the above circuit.

- D9 / PIN 9 – spare
- D10 / PIN 10 – PTT
- D11 / PIN 11 - FSK_PIN
- D12 / PIN 12 – CW

Default startup configuration

- nanoIO 1.0.1
- FSK: Baud: 45.45, Mark LOW
- CW: WPM: 18/18, dash/dot 3.00, incr 2, IambicA keyer
Computer Interface Command Set

USB serial command strings all begin with the tilde, ~, character

Cmd ~...

- C,c CW mode
- F,f FSK mode
- T,t CW Tune
- Snns computer wpm 10...100
- Unnu key (user) wpm 10...100
- Dnnd dash/dot 250...350 (2.5...3.5)
- In CW incr (1..9)
- A,a IambicA
- B,b IambicB
- K,k Straight key
- 0 FSK mark = HIGH
- 1 FSK mark = LOW
- 4 45.45 baud
- 5 50 baud
- 7 75 baud
- 9 100 baud
- ? Show config
- W Write EEPROM
- ~ Show cmds

In line special characters:

Both:

- [ PTT on
- ] PTT off
- \ escape; clear internal buffer and set PTT off

CW:

- ^ increase computer wpm by increment value (default is 2) *
- | decrease computer wpm by increment value *
- % <SK>
- & <AS>
- + <KN>
- - <BT>
- < <AS>
- = <BT>
- > <AR>
- { left brace <HM>
- } right brace <VE>

* - the ^| wpm modifiers do not effect the paddle CW, just the computer CW.
for example, a CW string might contain:

- \(~C~S24s[tu \textsuperscript{5}nn|| k]\)

1. Mode is changed to CW
2. Computer WPM is set to 24
3. PTT is set ON
4. The string “tu” is sent
5. Computer WPM is increased by 2 increment factors
6. The string “5nn” is sent
7. Computer WPM is decreased by 2 increment factors
8. The string “k” is sent
9. PTT is set OFF

PTT will enable before starting the CW transmission. PTT will disable after the last CW character (k) is completed.

nanoIO is a merger of code written by W1HKJ and:

tinyFSK, by Andrew T. Flowers K0SM
Iambic Keyer, by Steven T. Elliott

It expands the tinyFSK to include 100 WPM Baudot TTY. It shares a common set of control strings. The default wake up mode is FSK.

The keyer implementation is less robust than Steven's Iambic Keyer. It only provides Straight key, Iambic-A, and Iambic-B modes. It does add weighting to the key output. The key weighting and the computer CW weighting are the same. The current implementation does not have a tone output.

CW is generated from USB serial input strings and/or the paddle inputs. Paddle input overrides the serial string, but does not currently clear the string buffer. That could be changed, but the USB serial interface provides a very easy escape character to do just that.

**Configuring the sketch for target hardware.**

The nanoIO sketch can be used with one of four hardware designs

- MORTTY Version 2
- MORTTY Version 3
- HA2OS quad opto-isolator board
- W1HKJ quad opto-isolator board

MORTTY Version 2 shares a common output pin for both CW and FSK keyline. The user will have to swap FSK / CW line connectors when changing mode.

MORTTY Version 3 has separate output pins for CW, FSK and PTT keylines.

The W1HKJ and HA2OS designs also have separate CW, FSK and PTT keylines.

To select the h/w configuration when building the nanoIO sketch. Open the file config.h and uncomment the desired h/w design. The default is for MORTTY Version 3. A line is a comment if it begins with ‘//’.
// #define MORTTY-V2
#define MORTTY-V3
// #define HA2OS
// #define W1HKJ

//ifdef MORTTY-V2
    // Configuration for MORTTY Version 2 circuit board
    # define FSK_PIN 11
    # define CW_PIN  11
    # define PTT_PIN 13
    # define ST_Pin  4 // Sidetone Output Pin on Pin 4
    // paddle input pins compatible with MORTTY board
    # define LP_in   2 // Left Paddle Input on Pin 2
    # define RP_in   5 // Right Paddle Input on Pin 5
    # define DEFAULT_MODE CW_MODE
//endif

//ifdef MORTTY-V3
    // Configuration for MORTTY Version 3 circuit board
    # define FSK_PIN 12
    # define CW_PIN  12
    # define PTT_PIN 13
    # define ST_Pin  4 // Sidetone Output Pin on Pin 4
    // paddle input pins compatible with MORTTY board
    # define LP_in   2 // Left Paddle Input on Pin 2
    # define RP_in   5 // Right Paddle Input on Pin 5
    # define DEFAULT_MODE CW_MODE
//endif

//ifdef HA2OS
    // Configuration for HA2OS quad opto-isolator circuit board design
    # define FSK_PIN 12
    # define CW_PIN  10
    # define PTT_PIN 11
    # define ST_Pin  4 // Sidetone Output Pin on Pin 4
    // paddle input pins
    # define LP_in   5 // Left Paddle Input on Pin 2
    # define RP_in   2 // Right Paddle Input on Pin 5
    # define DEFAULT_MODE CW_MODE
//endif

//ifdef W1HKJ
    // Configuration for W1HKJ quad opto-isolator circuit board design
    # define FSK_PIN 12
    # define CW_PIN  11
    # define PTT_PIN 10
    # define ST_Pin  4 // Sidetone Output Pin on Pin 4
    // paddle input pins
    # define LP_in   2 // Left Paddle Input on Pin 2
    # define RP_in   5 // Right Paddle Input on Pin 5
    # define DEFAULT_MODE FSK_MODE
//endif