Alachua County NFARC Winkeyer Emulation

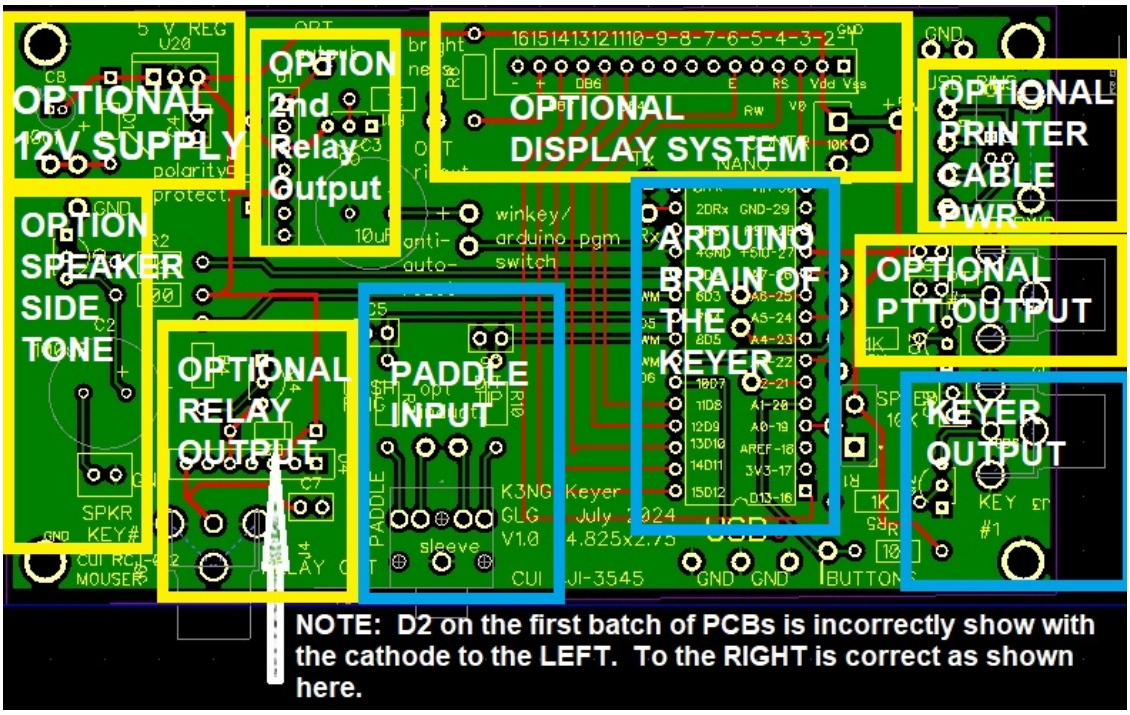
A LabNLunch Project!

August, 2024

Gordon Gibby KX4Z

Version 0.3

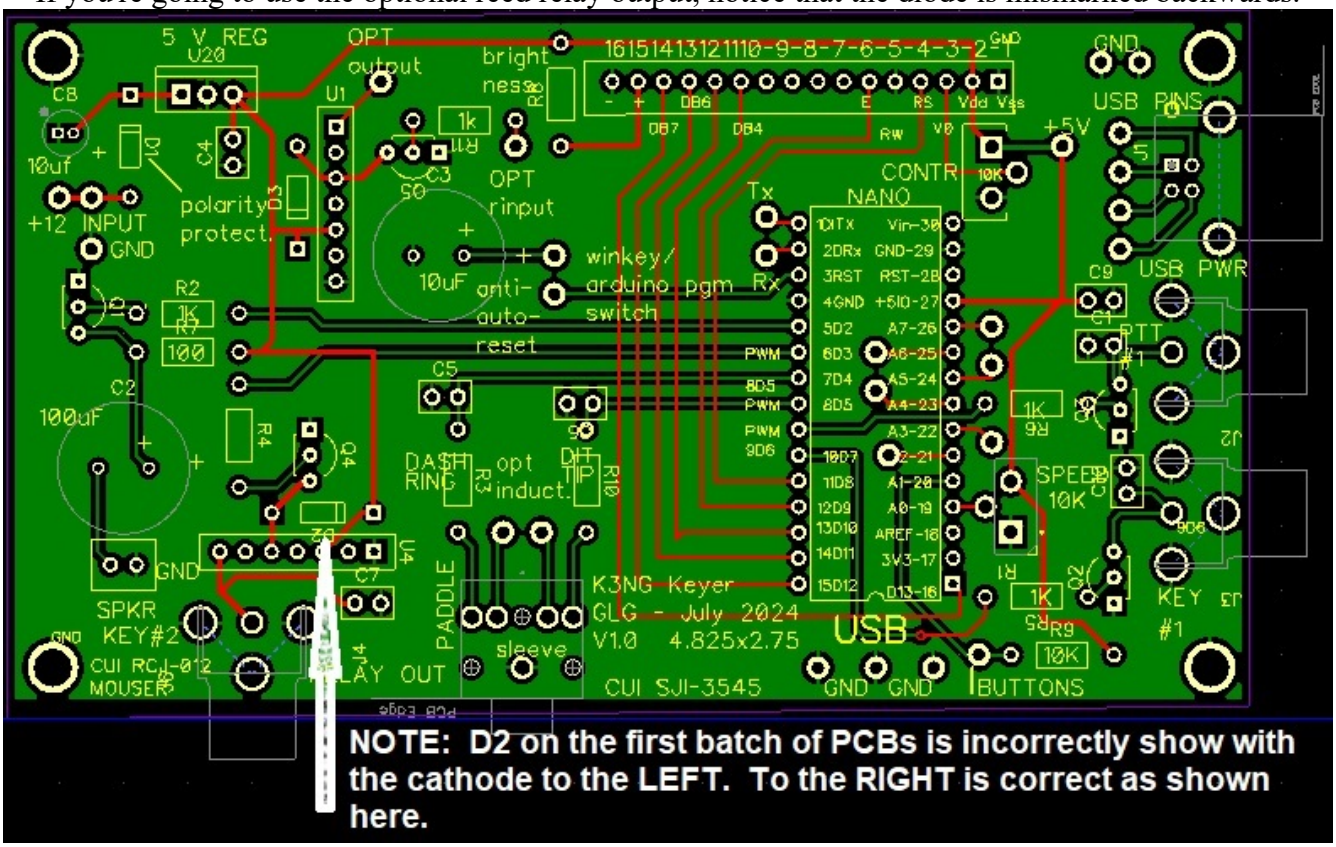
Manual Revision	Changes
0.1	Original
0.2	Corrected schematic to reflect 4K total resistors below 10K; added labels; added command label
0.3	Added explanation for how to modify relay output to avoid spot-welding due to the higher voltage switching (-55V) on Heathkit SB- HW- series transceivers



SECTIONS OF THE BOARD

CONTENTS

If you're going to use the optional reed relay output, notice that the diode is mismarked backwards.



FEATURES AND SPECIFICATIONS		
Stand-alone Electronic Keyer	Memory keyer	WINKEYER emulation
• Can be powered by either USB 5-volt source or 12-V source -- can be operated easily in the field with NO COMPUTER for POTA! • Simple TRS 3.5mm stereo paddle connections • Standard: Iambic A • Potentiometer easy speed adjust • 5-30 wpm base unit; can be expanded • Key line output open collector (2N3904) keys up to 10V, 20mA • PTT output open collector can key linear or transceiver with delays for relay actuation • Available reed-relay output for negative voltage, or high voltage (e.g. older vacuum tube transceiver) keying requirements • Available display shows word per minute setting and also shows all transmitted characters. • Speed can also be raised or lowered using command button and holding either dit or dash paddle • Does everything you need for a simple electronic keyer!	• Typical memory keyer construction has 3 memories, but this can also be expanded. • Memory text can be set to automatically repeat if desired; simple setting using paddle and individual button (hold button, touch dit paddle) • Repeat on/off is individually set for each memory text ("canned text") • Delay between repeats is also adjustable (same delay for any repeated) • If available display is installed, memory text is also displayed during transmit. • Memories are programmed using the paddle; no keyboard required. • Memories are maintained in non-volatile memory between instances of use.	• Provides Version 2.3 WINKEYER emulation • Works correctly with N3FJP • (Keying speed has been re-tuned to be primary with the keyer, not the computer for increased control.) • Standard 1200 baud WINKEYER emulation • Simply connect USB cable to the USB 2.0 Micro B jack on the Arduino Nano. • N3FJP provides for both canned text and direct keyboard to keyer sending as well (F12 key) • Will work with any logging system that keys using Winkeyer 2 protocol. • **Does everything you need to have a winning Field Day effort!**

OPERATION

NORMAL MODE

1. **STRAIGHT KEY MODE:** (rare) If you are holding the DASH arm pressed when you turn it on, it becomes a simple straight key (not a keyer) -- and the DOT arm is the Morse code straight key. Not sure when you would want this, but you can do it.

2. **SETTING KEYER SPEED:** When you normally turn it on, the Speed Potentiometer sets the KEYER speed. When you turn it, if there is a display, it will immediately start showing you your chosen speed.

3. **SENDING MEMORY TEXT:** If you press any of the MEMORY BUTTONS, it will start sending that memory canned text. It remembers them in non-volatile memory.

4. **REPEATING MEMORY TEXT:** If, while you press one of the MEMORY BUTTONS, you also press the DOT paddle, you have activated "repeat" and THAT memory will now repeat at the setting you assigned (see later). Repeating or not is unique to each memory. You have have whichever ones you want repeat.

COMMAND MODE

5. If you press the COMMAND button, you enter command mode. The screen will tell you that. I recommend a somewhat slower command speed so you don't make mistakes. But if you turn the potentiometer, you can change the speed of the command mode.

6. **EXITING COMMAND MODE**: If you press the COMMAND button again, or send X on the paddles, it will eXit command mode.

7. **USEFUL COMMAND MODE COMMANDS**: Other commands you can do while you are in COMMAND mode: (I selected what I thought were the most useful choices)

P1 <send the message> (fill the Memory #1)
P2 <send the message> (fill the Memory #2)
P3 <send the message> (fill the Memory #3)

Press the COMMAND button to terminate entry of the programmable memory text.
It will then send you back the message so you can hear what you have stored.
Press COMMAND again to get out of command mode

N -- causes the dot and dash paddles to swap sides. It is a "toggle" so you send N again to reverse back (remember, you swapped them, right?)
O -- cycles between three settings for the SIDETONE: ON, OFF, and PADDLE ONLY. It will show you on the screen

Y -- sets the time delay between memory repeats. 400 = 4 seconds, for example

T -- tune mode. This is very useful but also a bit confusing. Once you are in TUNE mode, you can hit COMMAND to exit, or press both paddles simultaneously. While in TUNE mode:

- Pressing the DASH arm starts a continuous transmit until you press it again
- Pressing the DOT arm acts like a Morse straight key-- it transmits as long as you hold it down.

NOTE: Exactly as your would want, the keyer does not key the transmitter while you are in command mode.

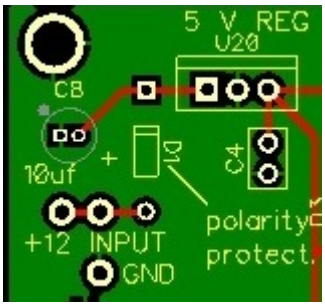
WINKEYER EMULATION

8. It also works with N3FJP in **Winkeyer emulation**. You plug a USB cable into the tiny connector right on the Arduino (Micro USB) . A short connector comes with the Arduino. You can get longer ones. I suspect it will work perfectly with N1MM also. I haven't tested Log4OM yet.

9. **Commands that were left out:** There are zillion OTHER commands but each one takes a bit of memory and I am at 96% full so far so not much much more can be added.....
It is an amazing keyer at this point. The USB-Winkeyer that I have was $140, works GREAT, but does not have a display. So this is really really cool for far less $$$$

10. **Display:** It is always showing you what you are typing, right on the display. Inside N3FJP, for example, you can hit F12 and then use the keyboard of your computer to "type" Morse code -- and the keyer will send it for you! Whatever letters you type, it sends--and shows you. Since we did NOT add a keyboard to the keyer, it doesn't have its own keyboard. Not enough memory left. But pretty amazing what you can do with this!

13.8VDC Power Source Option



It is recommended that if you are going to use this option, build it first of all, so that you can check its output voltage accuracy -- it should be between 4.75-5.25 VDC. BEFORE POWERING, use an ohmmeter or very careful visual inspection to be sure you don't have a short on the input! The pins of the 10uF capacitor are very close, as are the pins of the voltage regulator and the output 0.1 uF capacitor and a short at any of these can have "undesired" impact. The first time you apply power, start LOW. As the input voltage rises above about 8V, the output should stabilize right at 5 V. The input diode if correctly installed, will protect you against accidental reverse voltage.

This option allows you to power the keyer from any source of 8-20VDC. The board includes a series 1-ampere diode, so if you connect the polarity backwards, it just won't power the board. Regulation down to 5VDC required by the Arduino is provided by a 3-terminal linear regular, LM7805, which includes thermal protection. Because the current requirements should be low (provided excessive LED back light current isn't chosen) no heatsink is necessary.

When installing the wire into the Power Pole connector, the crimp won't be solid unless you fold back several thickness of the small wire so that some grip is obtained. Even then, it may not be suitable and I recommend CAREFULLY soldering from the BACK entrance -- any solder on the tang of the pin will make it impossible seat properly in the red or black plastic holder. You will need to use a small screwdriver or other implement to push forward to "seat" the pin in the plastic holder. If you've never done that, you may want to get help.

REQUIRED PARTS

Input connectors - power pole	Wire for input connection (#18 or #20, red/black)	D1 - 1N4007 Be sure to install with the banded end of the diode on the banded end of the printed circuit board screenprint.

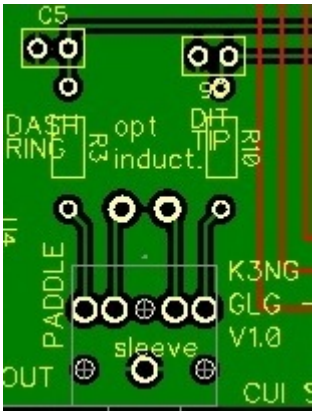
C8 10uF electrolytic Observe polarity when installing. The board is marked with + side, but many capacitors are only marked on the NEGATIVE side.	U20 5 Volt regulator Be certain that the metal tab is on the top side, as per the screen print.	C4 0.1 uF ceramic capacitor

Once the components have been installed, and SHORTS HAVE BEEN CORRECTED IF ANY, provide a voltage starting low as described above, and work your way up to 13.8VDC and test the output voltage.

THEORY OF OPERATION

The 7805 is a three terminal "linear" regulator that has an integrated circuit inside with some thermal and over current protection against shorts on the output. It can oscillate however, and for that reason input and output capacitors are used to reduce the feedback potentials from one end to the other of the device. If it is used for anything more than a few mA, it will need a small heatsink. The "05" at the end tells you is is for 5 volts. There are several other voltages available in the line of parts. There is also a "7905" that is intended to regulate NEGATIVE five volts, and a series there also.

PADDLE INPUT



The paddle input brings the switch closures from the paddle to digital inputs D4 and D5. These inputs are biased "high" with internal resistors on the Arduino.

REQUIRED PARTS

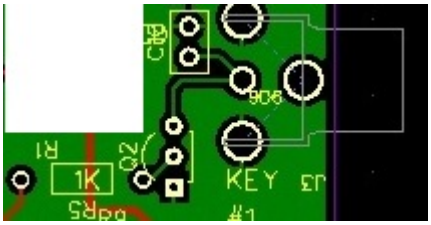
TRS Stereo 3.5mm jack for paddle input	Jumpers or inductors -- R3/R10 are place keepers for small inductors if you have severe problems with radio frequency interference. For now, install wire jumpers from cutoff portions of other computers or small sires.	0.01 uf bypass capacitors C5/C6. The markings on these capacitors are usually "103"

Theory of Operation

Software sets internal "pull up" resistors on these inputs of the Arduino so that they see "high" when the paddles aren't pushed. The paddle consists of two switches that can short a wire to ground. When one or the other paddle is pressed, the input to the Arduino sees "low" as long as that voltage drops below some threshold, typically 0.8 VDC.

The capacitors on the lines are there to prevent interference from radio frequency pickup. If these aren't sufficient, then add some inductance (e.g. 1 uH) in series where provided. Keep the wires to the paddle close together which provides some inherent shielding protection. In really bad situations, use shielded cable for this wire, and connect the shield to ground.

KEYED LINE OUTPUT #1



REQUIRED

This output allows the keyer to key positive-voltage keyed transceivers by grounding the key input with an open-collector output from a 2N3904. It cannot be used with negative voltage key lines, and certainly not with high negative voltage inputs like Heathkit vacuum tube transceivers from an earlier era -- use the reed relay output instead.

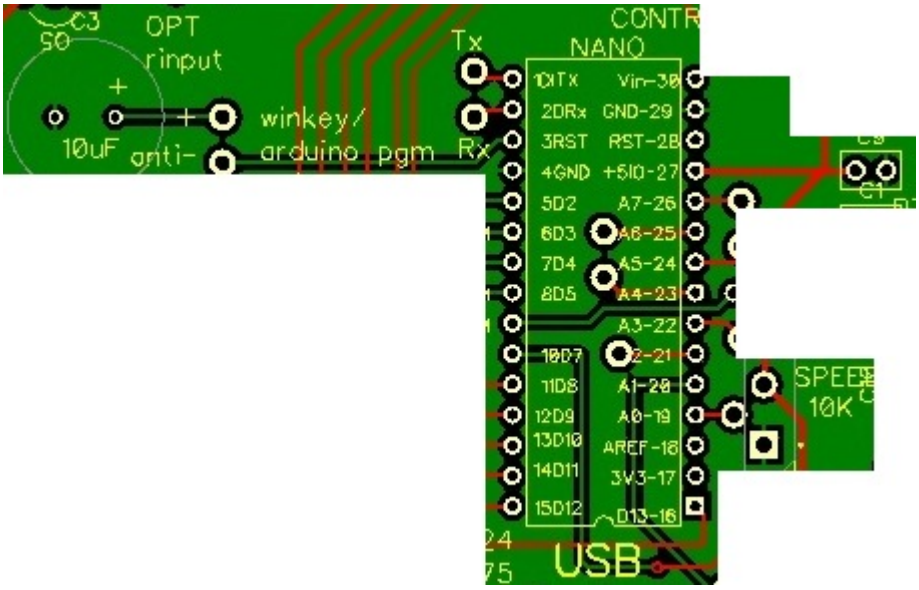
The maximum voltage and current that should be keyed are approximately +10VDC and 10 mA. Most transceivers will be well below this.

When the D7 output of the Arduino is driven high, 5 Volts begins to push current into the base of Q2 through the 1000 ohm resistor (approximately 4 mA; the Arduino may be able to push as much as 40mA but we don't want to overload it!). For normal keyed currents, this drives the transistor into full "saturation" and the collector voltage should be approximately 0.2 V, properly keying most transceivers

PARTS REQUIRED

RCA phono plug output connector J3	0.1 uf bypass capacitor	Q2 2N3904 transistor Be sure to install this with the flat end as shown on the screenprint. Avoid overheating the transistor. Maintain leads of about 2-4 mm between the transistor and the board, which will help with temperature protection.
R1 1000 ohms		

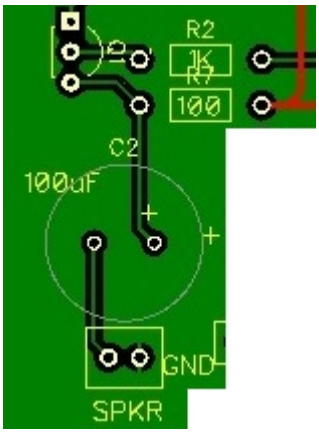
ARDUINO NANO



This part has 32 closely spaced pins. If yours comes without header pins, you'll need to solder the header pins to the Arduino and then the Arduino to the printed circuit board. You might want to use small spacers so the Arduino has a gap underneath it. That would make it easier to remove should you ever have to. Move quickly with the soldering iron to avoid overheating, and use just the required amount of thin rosin core solder to make a good connection. Don't leave a blob and be wary of any solder bridges. Use solder-wick to remove any bridges. Be certain the USB port is "down" as shown on the printed circuit board, and not "up"-

ARDUINO NANO		C9 0.1 uF bypass capacitor
C3 10uF anti-reset capacitor Once the Arduino is verified to work, this is connected in and prevents the Arduino from be auto-reset each time a command is sent to it over the serial port.	Wires or switch to chose whether programming reset is allowed -- we are going to per-program our Arduinos and just use a jumper wire here with a small loop so we could cut it if needed to reprogram.	10K Speed potentiometer (panel mounted potentiometer)

OPTIONAL SPEAKER OUTPUT

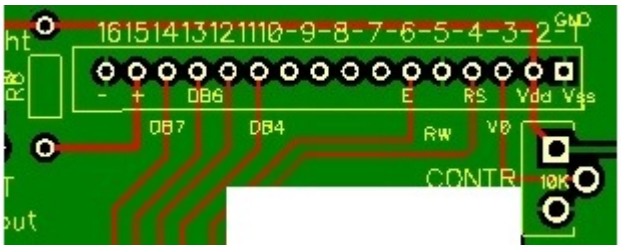


The speaker output is a square wave from the Arduino, amplified by transistor Q1 and AC-coupled to a small speaker by 100 uF capacitor C2. Be certain to observe polarity for C2.

The intensity of the volume is controlled by the choice of the value for R7. For loudest, pick 100 ohms. For fairly faint, pick 470 ohms.

Speaker - any small speaker. 8 ohms nominal impedance, but will work with almost anything. You can double-stick-tape this to the bottom of your box or plywood base	C2 100 uF	Q1 2N3904 Be careful to observe the placement of the flat side of the transistor.
C8 10uF electrolytic Observe polarity when installing	U20 5 Volt regulator Be certain that the metal tab is on the top side, as per the screen print.	C4 0.1 uF ceramic capacitor
R2 1000 ohms	R7 -- choose value based on sound intensity requested, from 1000 ohms (very faint) to 100 ohms (loudest)	Output terminals. If you prefer, you can put 0.1" header terminals here or otherwise just solder small wires to the speaker.

2x16 Character LCD Display



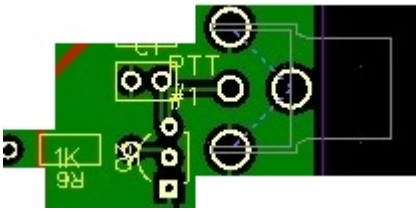
OPTIONAL
The LCD display must not be installed until AFTER the Arduino Nano, the contrast trimmer, 10uF C3, and portions of optional additional relay output (if installed) are soldered in, because they will be difficult to reach after the LCD is installed.

Be certain to recognize that the design places the display on the BACK side of the board, not on the Arduino side! Also be sure to use small spacer objects to keep the display off the board by 1/4" or more, so that it doesn't short out anything below it.

This is a standard inexpensive LCD alphanumeric display. For simplicity, in this design it is connected for 4bit (nibble) data. An alternative design might be to control it using I2C, but this was not chosen for this design. The display includes a backlight LED system, which must be fed a modest amount of current using dropping resistor R6 just to the left of the display. That resistor can be 470 ohms for modest lighting, all the way to 100 ohms for brighter.

LCD display		Screen brightness resistor
Contrast trimmer (10K)		

OPTIONAL PTT Radio Control



This isn't necessary, but it can be helpful for really fast paced contest contacts, because it can allow you to "un-transmit" faster than waiting for the VOX delay to expire when the last letter is transmitted. So I always use it in contests.

RCA Phono PTT output connector	Q3 2N3904 Transistor	C1 0.1 uf Capacitor
R6 1K resistor		

Theory of Operation
When the Arduino output goes high, it drives current through the 1K ohm resistor into the base of the transistor, pushing the transistor into full conduction (called "saturated"). Most transceivers provide a small positive voltage on their input and shorting this to ground with the saturated collector will push the radio into "transmit" mode. The normal delay going back to "receive" may be eliminated when this transistor goes quickly into non-conduction.

BE CAREFUL WITH OLDER TRANSCEIVERS -- Vacuum tube radios frequently used very large NEGATIVE voltages on these connections for push to talk, and that will destroy circuitry on this board! Use a relay output instead of a transistor output if you need to control push to talk on a vacuum tube older radio. Check that your radio does not source more than single digit milliamperes through its push to talk.

SPEED POTENTIOMETER

Very useful -- much much faster than working through menus.
10K potentiometer, connected by wires to the board.
The ones I normally use require a 0.375 (3/8") hole and a second hole for the indexing pin.

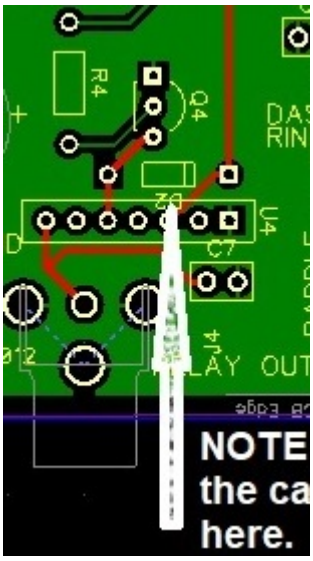
BUTTONS

10K resistor from +5 gets shorted to ground by Command button.
1 K ohm resistor from the 10K goes to the Memory #1 button, which can short to ground
1 K ohm resistor next goes to the Memory #2 button, the other side of which is also connected to ground.
1 K ohm resistor goes to the Memory 3 button, and then a 1 K resistor from there to ground.

Depending on which button shorts to ground, a different voltage shows up at the bottom of the 10K. It is zero when the command button is pushed, slightly greater when Memory #1 is pushed, again greater if Memory 2 is pushed and so on.

The buttons can be positioned pretty much anywhere, but the side of the box works well.

OPTIONAL RELAY KEY LINE OUTPUT



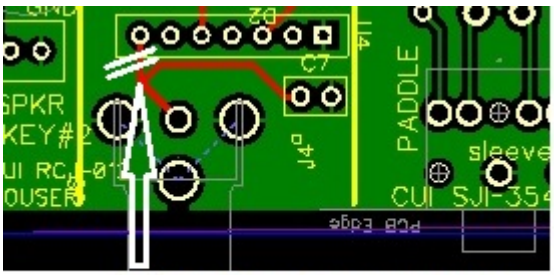
The optional relay keyed line output is as laid out, connected to "key output #2". Our software doesn't normally send to this output, so Resistor R4 can instead be diverted so that its top side comes from the Arduino output that feeds Key Line #1, allowing both open collector and the relay system to be keyed simultaneously. Otherwise, arrange for special programming of your unit to allow access to Key output #2.

An esrlier error has been corrected. Diode D2 is properly shown with the cthode to the left. .

Install the components including the transistor, diode, resistor and the 4 pin reed relay. The relay can go either way, but pin #1 is the square pin if you want to be exact; the part is symmetrical so it doesn't matter. Also add C7 the bypass capacitor and the phono socket output.

CORRECTION to AVOID SPOT-WELDING

It was found that the high voltage (-55V DC) of Heathkit SB-100 / HW-100 series transceivers caused the delicate contacts of the reed-relay to "spot weld" closed. To aovid this it is strongly recommended that the top trace (shown in red) above from the phono-socket to pin 1 of the relay be scratched thrugh, and a 2.2k ohm resistor used to connect the center pin of the phono jack to the relay. This llimits the peak current handled by the relay and appears to pose no change in operation of the older transceiver. Additional the bypass capacitor should be moved to the output connecto side of the resistor, NOT across the contacts of the relay, and may be reduced to 0.001 uF



Cut PCB trace here and bridge with 2200 ohm resistor

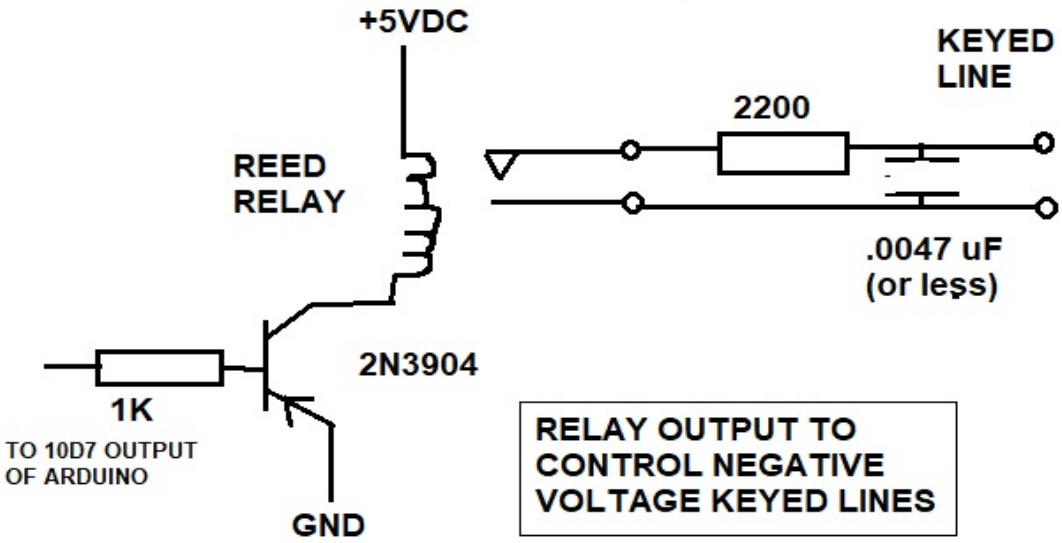


Figure: corrected relay output schematic to avoid spot-welding on the fragile reeed relay contacts by -55VDC from Heathkit HW-100 SB-100 series transmitters

CODING NOTES

Subroutine
void command_mode()

Commands Implemented from CW at the paddle:

N - reverse paddles
O - cycle through side tone modes (off, paddles only, etc)
P - program memories (next digit is the number of the memory you program; then send text and end with closing command mode by pressing command button again)
T - set tuning mode
Y - set repeat delay [applies to all memory once turned on]
Unknown command - implemented (sends ?)

COMMAND LABEL (consider placing on bottom of device)

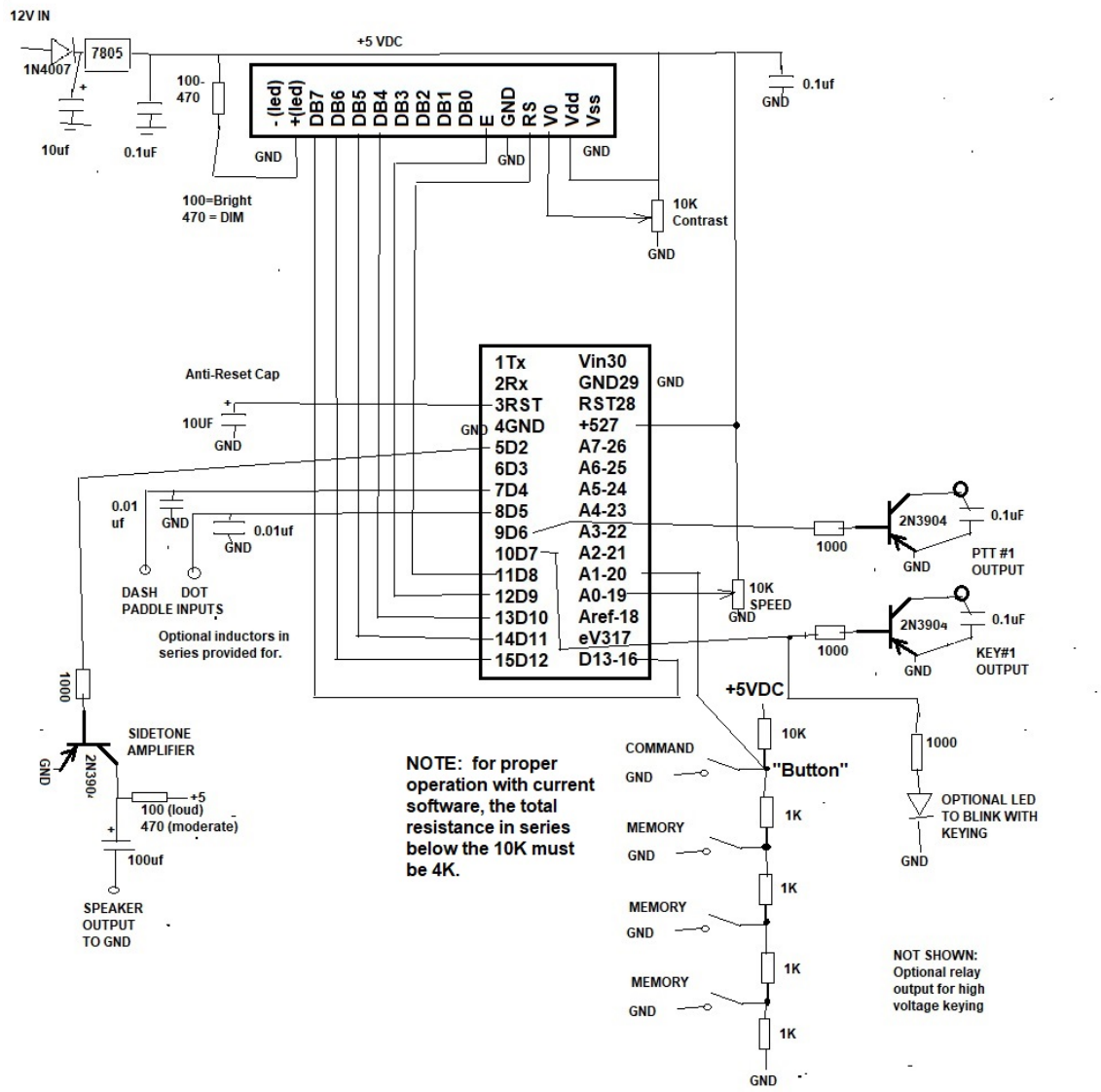
CW Command	Action
N	Reverse paddles
O	Cycle side tones (off/on/paddles)
P	P1 = program #1 P2 = program #2 etc
T	Tuning mode (left/right paddle)
Y	Set repeat delay <add number> (Hold button, touch dit paddle to order repetitive)

Commands at the paddle not implemented

A - iambic mode (default, not implemented)Un
B - B iambic mode (command not implemented)
C - single paddle mode (command not implemented)
E - announce speed (command not implemented)
D - Ultimatic mode (command not implemented)
G - bug mode (command not implemented)
H - weighting adjust (command not implemented)
I - toggle transmit (command not implemented)
J - adjust dit to dah (command not implemented)
L - weighting adjust (command not implemented)
M - command mode speed (set to 10) (command not implemented)
Q - set keying compensation (command not implemented)
R - set serial number (command not implemented)
V - toggle speed pot active (command not implemented)
W - change speed (command not implemented) (use speed potentiometer)

Code uses 96% of available program space; 61% of available RAM

SCHEMATIC
(corrected, 9/22/2024)\
(see earlier for relay output change schematic)



COMPONENTS REQUIRED

No.	Item	Quantity (max)	Comment
1	Arduino Nano	1	
2	2x16 LCD display	1	
3	RCA phono jacks	3	minimum 1, maximum 3
4	TRS stereo paddle jack	1	
5	10K trimmer potentiometer	1	contrast
6	10K panel potentiometer	1	speed (Req 0.375 drilled hole)
7	Momentary contact pushbutton	4	
8	Power Pole	1 pair	
9	7805 voltage regulator	1	optional
10	2N3904 transistor	4	2 are optional
11	Diode	2	one for 12V power, 1 optionalfor reed relay
12	LED	1	optional; use 1K dropping resistor
13	1 K resistor	8	some optional with buttons
14	0.1 uf Capacitor ceramic	5	
15	0.01 uf capacitor	2	paddle terminals
16	10 uF electrolytic capacitor	2	no-reset, and 12V power input (opt)
17	100-500 ohm LED resistor	2	speaker and back light
18	100 uf capacitor	1	for speaker output; optional
19	small speaker	1	optional
20	10K resistor for button	1	top resistor
21	Plastic box	1	
22	Standoffs	4	
23	3mm screws/nuts	4ea	
24	16 pin header	1	
	OPTIONAL Components for Keyed-Line Relay Output		

	5V reed relay	1	OPTIONAL
	2200 ohm resistor	1	In series with relay contacts
	0.001 or 0.0047 cap	1	RF bypass, possibly unnecessary

APPROXIMATE COST, AUGUST 2024

Select	INCLUDES	CAPABILITIES	INCRE-MENTAL CHARGE
❑	• BASE unit mounted open on wood • Printed circuit board • Arduino Nano (https://www.amazon.com/gp/product/B07G99NNXL) • Key & PTT 2N3904 outputs • USB short cable to Nano • One command button • Speed Potentiometer • No Display • No Box (case) • No Relay output YOU provide tiny speaker (e.g. from Baofeng mic) YOU provide paddle system with TRS 3.5 mm stereo plug to connect to provided TRS 3.5 mm JACK on printed circuit board.	• Fully functional iambic keyer with TRS stereo jack for paddle input • Speed adjustable from 10-30 wpm • Command speed set to 10 wpm • Capable of keying most modern solid state transceivers via its open-collector 2N3904 outputs (~up to 10V, 20mA) • Can control push-to-talk also • (Build your keying cable from stereo cable and either 1/4" or 1/8" plug as needed for your radio, not included here.) • Fully functional WINKEYER connection to N3FJP (tested) and likely most other modern logging systems. • Keying speed controlled primarily via Potentiometer, but can also be controlled by paddle (hold command key down, press left paddle to speed up, right paddle to slow down)	Board $5.33 Connectors $6.00 Transistors $0.60 Resistors $0.60 Electrolytics $0.50 Ceramics $0.50 Button $0.53 Arduino Nano $5.00 4 standoffs $0.56 TOTAL $20.72
❑	• Add Small speaker	• (Option provided in case you don't have one) -- you are likely to want the side tone • Something similar to this: • https://www.amazon.com/Metal-Shell-Internal-Magnet-Speaker/dp/B0B5KZL868	Adds $1.25
❑	• Add three buttons for Memories • Add required Resistors for buttons	• Allows up to three canned texts to be programmed and/or initiated, and each can be individually set to "repeat"; the repeat time can be set (but not	Adds $2

		individually)	
❑	• Add plastic box	• Nice box from Amazon • Drill points copied from my prototype • https://www.amazon.com/Raculety-Waterproof-Electrical-Electronic-Enclosure/dp/B089K1KBL6	Adds $7.50 plus you get the experience of DRILLING
❑	• Add 2-line LCD display and header and trimmer potentiometer	• The display shows the speed when you change it, an d always shows the code being transmitted as well. Makes programming memories a little more obvious also. • Example: https://www.amazon.com/gp/product/B00HJ6AFW6 • Header from stock -- • https://www.amazon.com/gp/product/B09G5RF3H6	Adds $5.00 plus you get the experience of DRILLING and SAWING
❑	• Add reed-relay output for radios with negative or high voltage keying voltage (e.g. Heathkits)	• Relay similar to • https://www.amazon.com/HiLetgo-SIP-1A05-Switch-Resistance-450-550ohm/dp/B09KGYPWPP • plus 2N3904 and a few small components	Adds $3
❑	• Add little LED to flash when keying	• Just so you can see it if you want.	Adds $0.50
❑	• Add 12V input power option with three terminal regular and power pole connector assembly	• Allows keyer to operate from 12V without need for any computer • L7805 regulator, a few capacitors, and power pole • https://www.amazon.com/outstanding-L7805CV-Regulator-Positive-Three-Terminal/dp/B08HGDG6WB	Adds $4 and you get the experience of crimping power poles (use multiple folded of small wires to be sure it crimps properly)

		• Genuine Anderson power pole connectors: • https://www.amazon.com/Anderson-PP15-12-16-AWG-Set/dp/B07BKSXF4G	
	You add up the total for all the options YOU would like:		TOTAL

LABELS

RELAY OUTPUT ARDUINO WINKEYER Aug 2024							
ADUINO WINKEYER **OPEN COLLECTOR OUTPUT**					LOW POS DC ONLY		
					PTT OUT	KEY OUT	
		Memory1	**Memory 2**	**Memory3**			
Arduino Micro USB		**Paddle Input**	**Relay Output**	**Speed Control**			
Command							
COMMAND							
						Powerpole 8-20 VDC	
ARDUINO WINKEYER **RELAY + OPEN COLLECTOR OUTPUT** **Aug 2024 KX4Z**							