;USKI-5 PROGRAM (D) PROVIDES SEPARATE CW AND PTT INPUTS AND OUTPUTS WITH "SEND" MONITORING FROM RIG AND INHIBIT CAPABILITY. CW TRANSMITTION IS ONLY POSSIBLE WITH PTT CONTROL AND NOT FOR SEMI OR FULL BREAK IN.

;REFER TO THE USKI-5 USER MANUAL FOR PROPER INPUT AND OUTPUT CONNECTIONS TO THE SEQUENCER.

;ALL DELAYS ARE IN MILLISECONDS (ms)

symbol TX\_START\_DEL\_1 = 8 \* 0 ;output 1 start delay, normally zero,

symbol TX\_START\_DEL\_2 = 8 \* 100 ;output 2 start delay

symbol TX\_START\_DEL\_3 = 8 \* 100 ;output 3 start delay

symbol TX\_START\_DEL\_4 = 8 \* 100 ;output 4 start delay

symbol TX\_START\_DEL\_5 = 8 \* 100 ;output 5 start delay

symbol TX\_END\_DEL\_5 = 8 \* 100 ;output 5 end delay,

symbol TX\_END\_DEL\_4 = 8 \* 100 ;output 4 end delay

symbol TX\_END\_DEL\_3 = 8 \* 100 ;output 3 end delay

symbol TX\_END\_DEL\_2 = 8 \* 100 ;output 2 end delay

symbol TX\_END\_DEL\_1 = 8 \* 100 ;output 1 end delay

pullup on ;turns on pullup resistors for all inputs

setfreq M32 ;overclock to 32 MHz

main:

initialize: ;initialize outputs upon power up. low = OPEN, high = CLOSED

low B.1

low B.2

low B.3

low B.4

low B.5

standby: ;waits in this loop until cw or ptt input

setint %00000000,%00000100 ;sets interrupt(inhibit)on input C.2 (J2R)

if pinC.2 = 0 then goto standby ;checks for interrupt on C.2(J2R closed)

bit0 = 0 ;clears any interrupt flag, if necessary

if pinC.0 = 0 then goto pttstart ;checks for PTT input C.0(J1R closed)

if pinC.3 = 0 then goto safe ;checks for accidental direct keying of rig

goto standby

safe: ;run if xmtr/xcvr accidentally keyed or PTT directly

high B.1

high B.2

high B.3

if pinC.3 = 0 then goto safe

pause TX\_END\_DEL\_4

low B.4

pause TX\_END\_DEL\_3

low B.3

pause TX\_END\_DEL\_2

low B.2

pause TX\_END\_DEL\_1

low B.1

goto standby

pttstart: ;beginning of PTT TX start procedure

setint %00000000,%00000100 ;resets interrupt, if necessary

pause TX\_START\_DEL\_1 ;output 1 start delay

if pinC.0 = 1 then goto standby

if bit0 = 1 then goto standby ;checks if interrupt occurred during pause 1, if so goes back to standby

high B.1 ;switches output 1

pause TX\_START\_DEL\_2 ;output 2 start delay

if pinC.0 = 1 then goto pttend1

if bit0 = 1 then goto pttend1 ;checks if interrupt occurred during pause 2

high B.2 ;switches output 2

pause TX\_START\_DEL\_3 ;output 3 start delay

if pinC.0 = 1 then goto pttend2

if bit0 = 1 then goto pttend2 ;checks if interrupt occurred during pause 3

high B.3 ;switches output 3

pause TX\_START\_DEL\_4 ;output 4 start delay

if pinC.0 = 1 then goto pttend4

if bit0 = 1 then goto pttend4 ;checks if interrupt occurred during pause 4

high B.4 ;switches output 4

pause TX\_START\_DEL\_5

if pinC.0 = 1 then goto pttend5

if bit0 = 1 then goto pttend5

ptthold: ;holds in this loop while PTT on

setint %00000000,%00000100 ;sets interrupt(inhibit)on input C.2 (J2R)

if bit0 = 1 then goto pttend5 ;checks for interrupt

if pinC.1 = 0 then goto cwloop

if pinC.0 = 0 then goto ptthold ;loops while PTT input continues

pause TX\_END\_DEL\_4 ;output 4 end delay

cwloop:

if bit0 = 1 then goto pttend5

high B.5

if pinC.0 = 1 then goto pttend5

if pinC.1 = 0 then goto cwloop

low B.5

goto ptthold

pttend5:

low B.5

pause TX\_END\_DEL\_4 ;output 5 excluded, only used by CW routine in this program

pttend4:

low B.4 ;unswitches output 4

if pinC.3 = 0 then goto pttend4 ;checks for release of "send" from xmtr/xcvr

pause TX\_END\_DEL\_3 ;output 3 end delay

pttend3:

low B.3 ;unswitches output 3

pause TX\_END\_DEL\_2 ;output 2 end delay

pttend2:

low B.2 ;unswitches output 2

pause TX\_END\_DEL\_1 ;output 1 end delay

pttend1:

low B.1 ;unswitches output 1

goto standby ;go back to standby

interrupt: ;interrupt routine

bit0 = 1 ;set interrupt flag on

return ;return to point where interrupt occurred