

Cool Microcontroller Projects

ICOM/PC Interface Controller

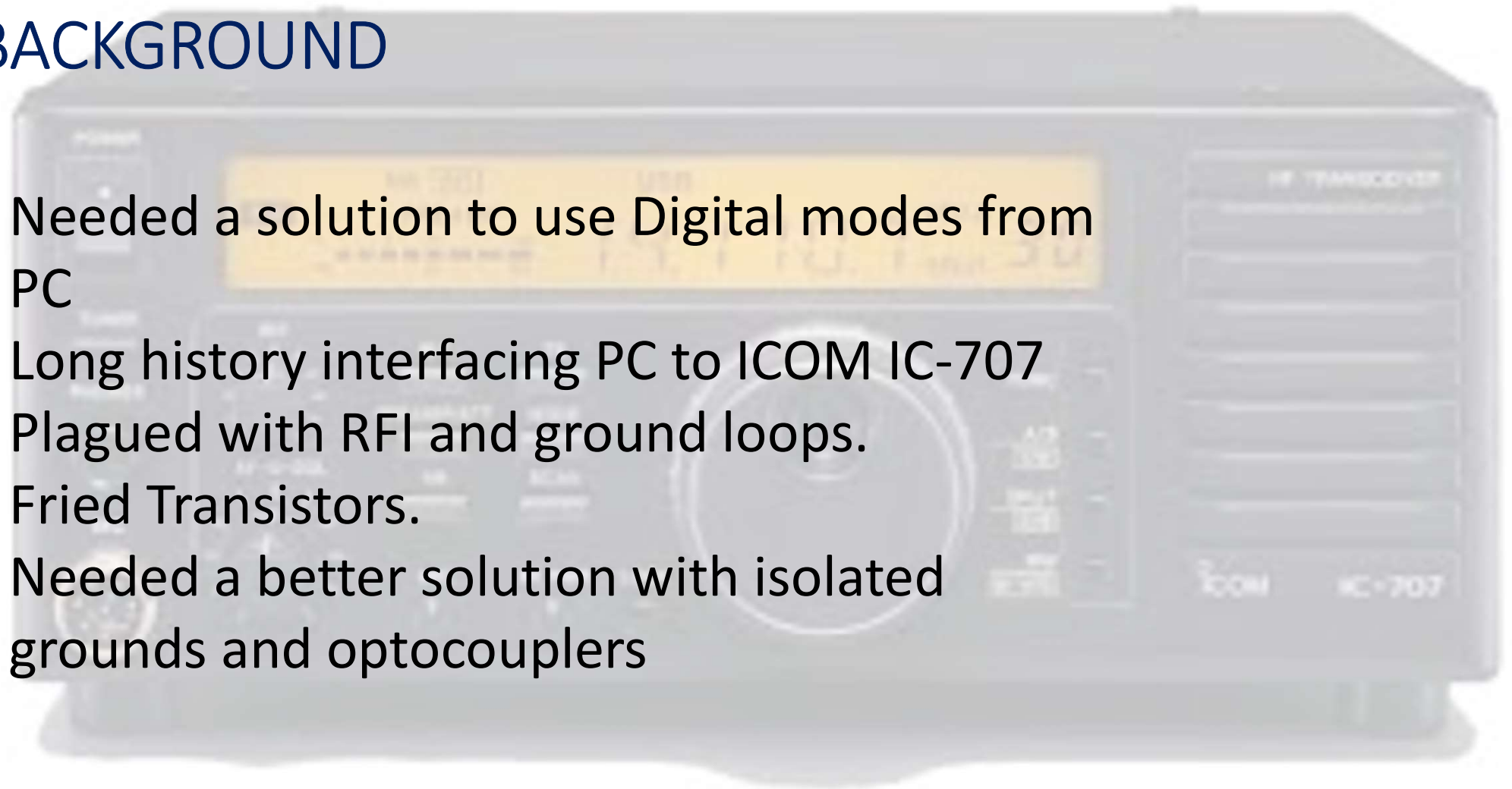
Dave VE3OOI

Feb 2021

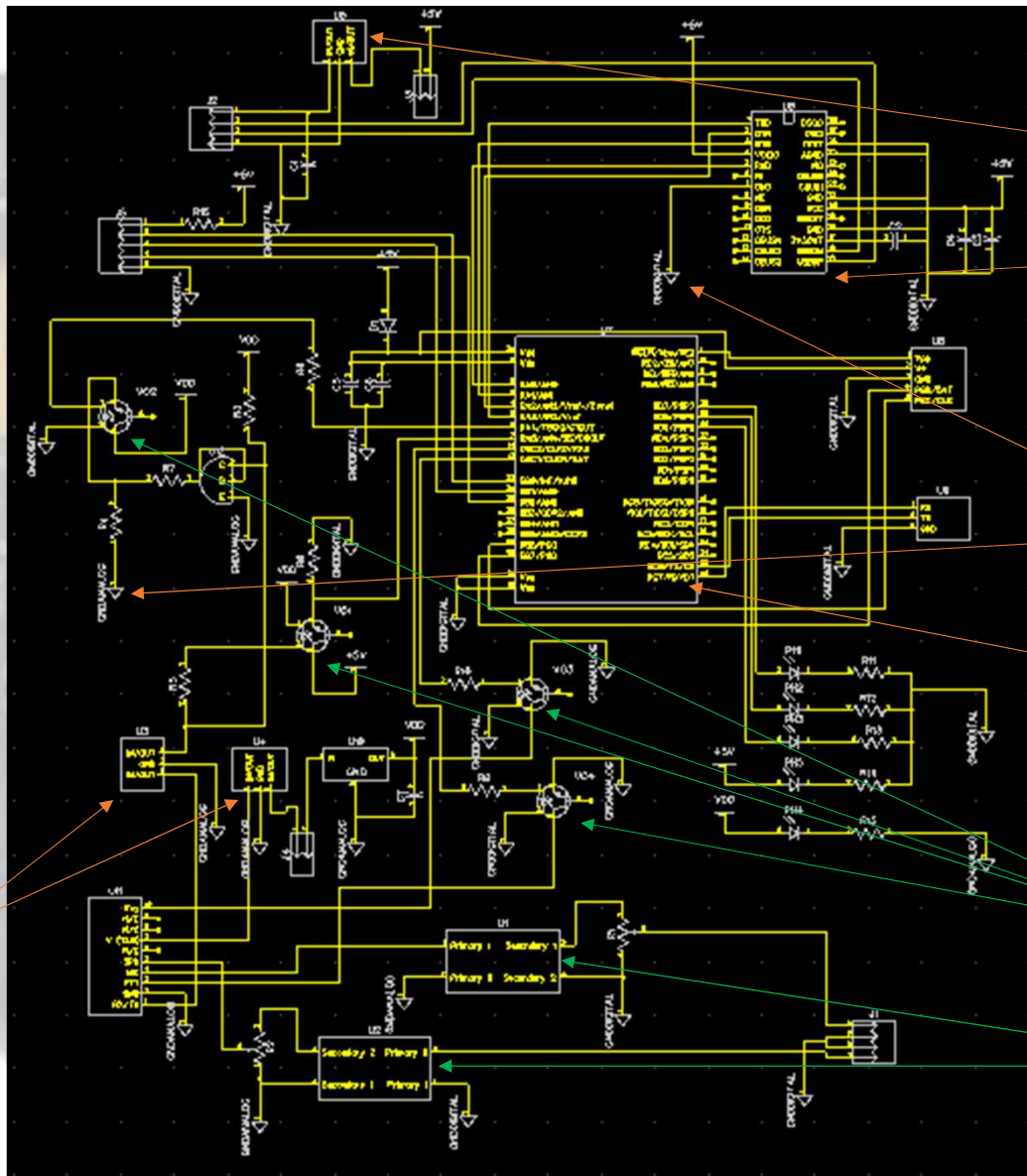


BACKGROUND

- Needed a solution to use Digital modes from PC
- Long history interfacing PC to ICOM IC-707
- Plagued with RFI and ground loops.
- Fried Transistors.
- Needed a better solution with isolated grounds and optocouplers



CIRCUIT



USB Power Supply Choke

USB Interface Chip

Two separate grounds that NEVER touch

PIC 16F777

Optoisolators

Audio Transformers

Radio Power Supply Choke

PCB

USB PS Switch

USB Interface

RF PS Switch

Radio Interface

ICOM Inverface
V1.0 VE300I

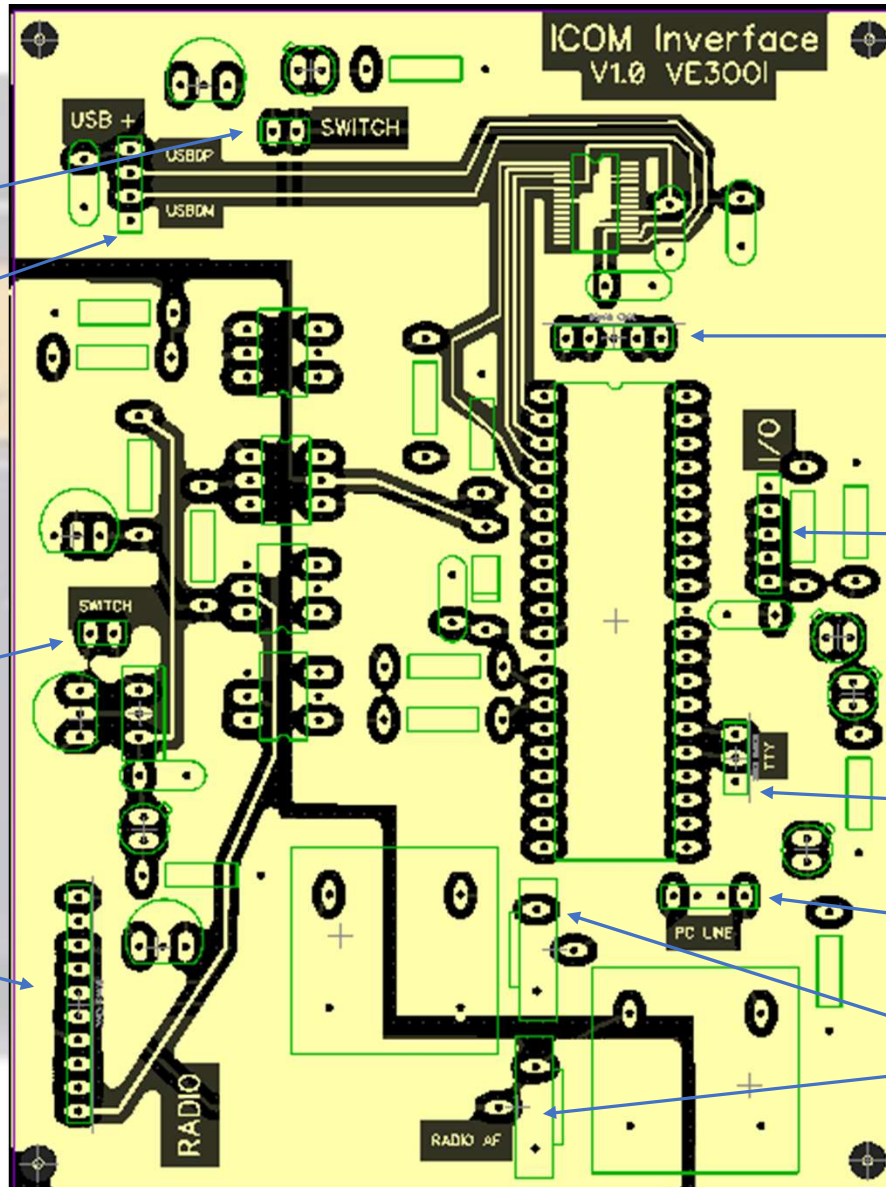
ISCP Interface

Future Interface

TTY Interface

PC Sound Card Interface

Sound Level POTs

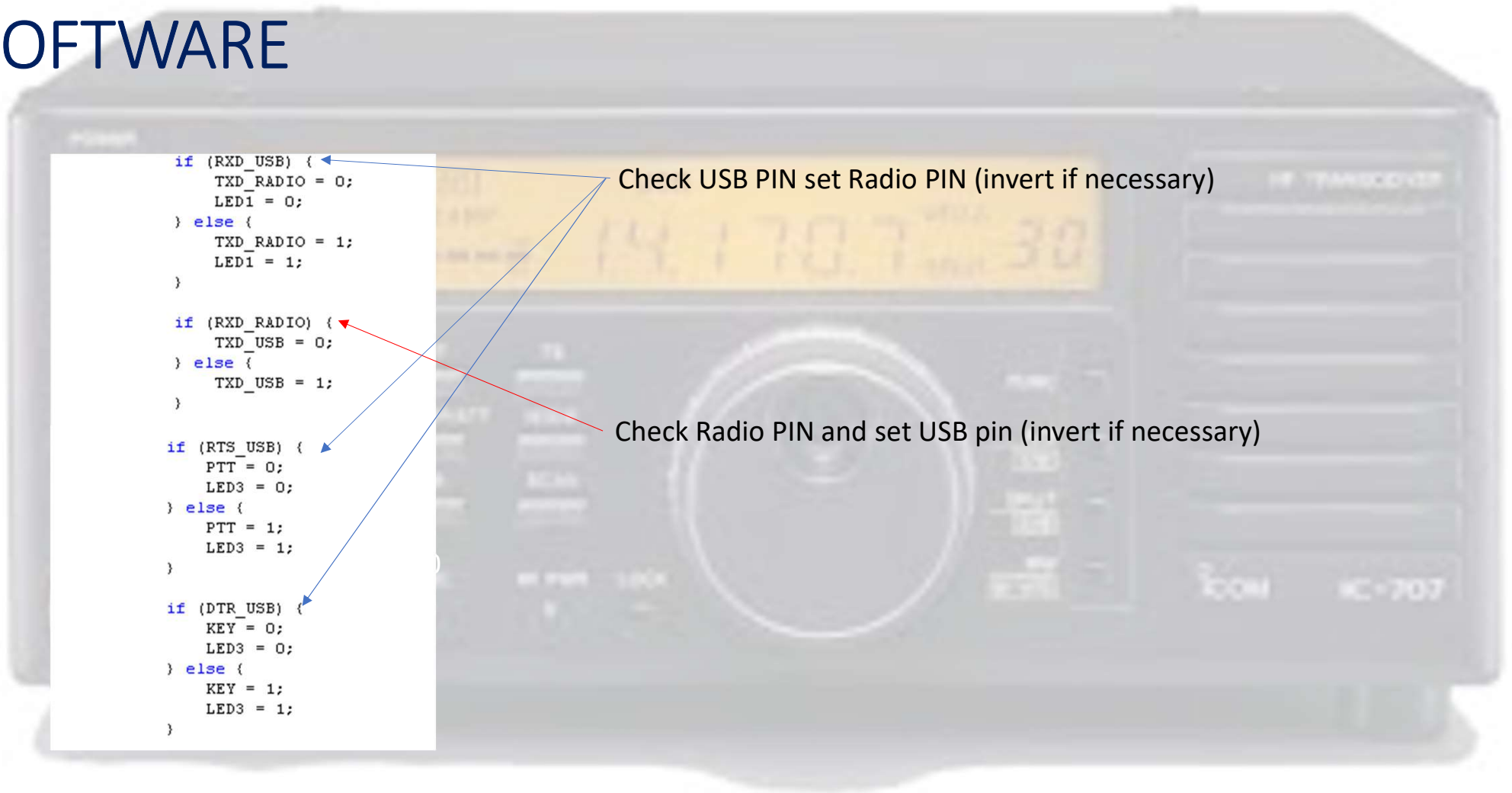


SOFTWARE

```
if (RXD_USB) {  
    TXD_RADIO = 0;  
    LED1 = 0;  
} else {  
    TXD_RADIO = 1;  
    LED1 = 1;  
}  
  
if (RXD_RADIO) {  
    TXD_USB = 0;  
} else {  
    TXD_USB = 1;  
}  
  
if (RTS_USB) {  
    PTT = 0;  
    LED3 = 0;  
} else {  
    PTT = 1;  
    LED3 = 1;  
}  
  
if (DTR_USB) {  
    KEY = 0;  
    LED3 = 0;  
} else {  
    KEY = 1;  
    LED3 = 1;  
}
```

Check USB PIN set Radio PIN (invert if necessary)

Check Radio PIN and set USB pin (invert if necessary)



ISD1750

```
void VcoderFunction (char op)
{
    switch (op) {

        case 'R':                                // Start Recording
            REC = 0;
            delay_ms(30);                        //24 ms Debounce time
            break;

        case 'S':                                // Stop Recording
            REC = 1;
            delay_ms(30);                        //24 ms Debounce time
            break;

        case 'P':                                // Play
            PLAY = 0;
            delay_ms(30);                        //24 ms Debounce time
            PLAY = 1;
            while (!RDY); ←
            break;

        case 'F':                                //Forward - 13 Spots
            FWD = 0;
            delay_ms(30);                        //24 ms Debounce time
            FWD = 1;
            break;

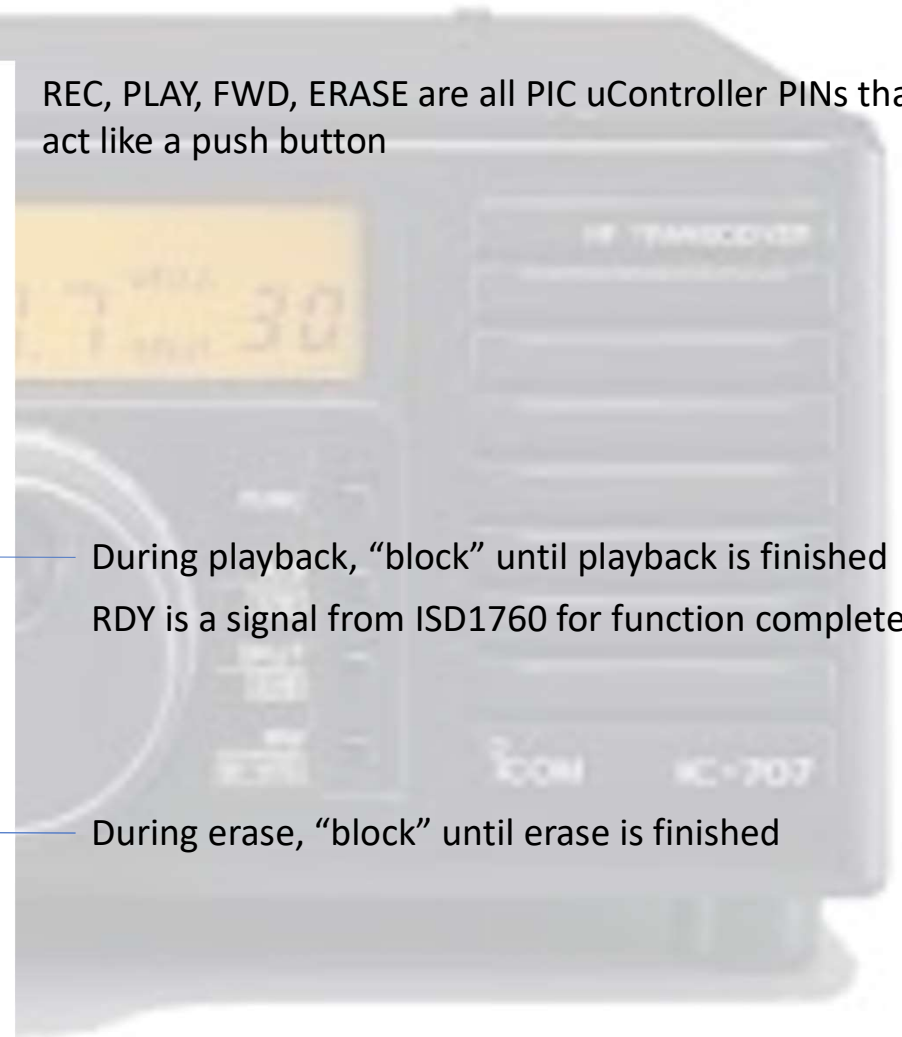
        case 'E':                                //Current Erase
            ERASE = 0;
            delay_ms(30); ←
            ERASE = 1;
            break;

        case 'B':                                //Global Erase - Blank
            ERASE = 0;
            delay_ms(30);
            while (!RDY);
            ERASE = 1;
            break;
    }
}
```

REC, PLAY, FWD, ERASE are all PIC uController PINs that act like a push button

During playback, “block” until playback is finished
RDY is a signal from ISD1760 for function complete

During erase, “block” until erase is finished



BUILD



