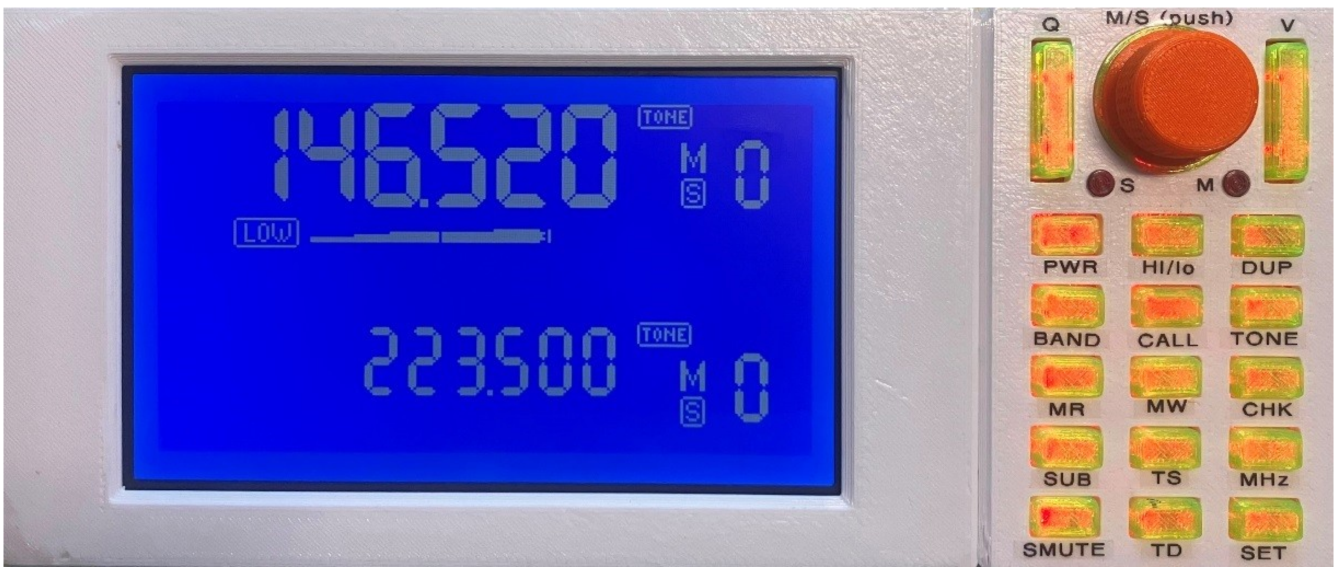
IC-900F Remote Controller
Theory Of Operation

Joe Haas, KE0FF
10/27/24



Introduction

This document describes the design of the IC-900F, an alternate Remote Controller design for the IC-900 radio. Particular focus is on the new hardware and software aspects of the design. References to the mechanical and electrical elements peculiar to the original IC-900 system may also discussed, and the operator is referred to the IC-900A Instruction Manual, published by ICOM for more detail regarding the original radio design.

This project was born by an intersection of chance and whimsy to create an alternate controller for the IC-900. The design followed a progression of steps that led to a modular approach which resulted not from a conscious choice, but more from serendipitous roots. The original focus was to produce a replacement controller which utilized the original LCD, LCD drivers, and enclosure. From this result, it was decided that a replacement display could be constructed, one which used the same protocol as the original LCD drivers. This was aimed at an all-from-scratch design that would produce a controller that did not depend on any of the parts from an original control unit.

The result is somewhat inefficient in terms of microcontroller utilization, but the advantage is that the modular nature of the design makes it easier to adapt to different use-cases. The display can be used with an original controller to produce a replacement or auxiliary display – the control board can be deployed alone to augment or repair an original controller, or the entire ensemble can be deployed together as a completely new controller unit.

It should be noted that the information in this document focuses on the circuits produced as part of this project. This design was, in large part, instructed by the contents of the ICOM IC-900 Service Manual, which details the operation of the signals and circuits which pertain to this design. Consult the service manual for more information regarding these signals.

Contact information

The author/designer may be contacted at the following:

Joe Haas, KE0FF
email: joeh@ke0ff.org
web: www.ke0ff.org
github: https://www.github.com/ke0ff/

The Controller Board

The controller circuit board consists of a Tiva TM4C123GHPM microcontroller (MCU), a power supply, a data separator for the A-B Unit communications connection, and circuits for the backlight and button interface. The form-factor of the PCB matches that of the original ICOM PCB which allows it to be a drop-in replacement. *Note: the "_33" suffix on net names differentiates signals that are translated to/from 5V logic (there is no suffix for the 5V logic signals, nor is there a suffix for 3.3V logic signals that are not translated to/from 5V. HCT logic family devices are used to convert 3.3V logic to 5V (output). For inputs, a resistive divider is used to convert 5V logic to 3.3V.*

MCU

The MCU circuit contains the memory and I/O resources for the design. An add-on NVRAM is used to contain the radio state and memory data. The size of this memory allows for approximately 300 memories for up to 6 different bands. ICs U1, U16, U17, and associated components (resistors, capacitors, and crystals) form the core of the MCU. Most of the General Purpose I/O (GPIO) of U1 is dedicated to decoding the button matrix. Other I/O is used to communicate with the A-Unit (SIN and SOUT), the user UART command-line interface (CLI), Serial connections for the display interface and the NVRAM device, an ADC input to measure ambient light conditions, and PWM outputs to control LED brightness. A serial UART to communicate with a Bluetooth module is also provided as well as provisions for that module, but this interface is not currently utilized.

There are two classes of button inputs: the 2x9 matrix (18 buttons) and the direct input buttons (3 buttons, DIM, CHECK, SMUTES4). For the 2x9 matrix there are two column select outputs, COL0 and COL1. These are complemented in software so there really is just one column select signal that is split into a true and a complement. The COLx outputs are used to excite the desired column on the external interface board. Software continuously toggles between COL0 and COL1 and the software captures all 9 inputs at once (PA[7:2], PB4, and PE[3:2]) and applies debounce techniques. The SW also differentiates between "press", "hold", and "release" and passes this information to the executive for processing.

The system uses an "up/down" rotary encoder (DIAL_UP and DIAL_DN) which the MCU captures using the ARM GPIO edge-detect interrupts. The system software was modified to also accommodate a quadrature encoder input since "up/down" encoders are more difficult to source for the clone implementation.

The NVRAM and LCD interface both share the same SSI peripheral (also called SPI). The system software uses MISO_LOCK, MOSI_33, and /SCK_33 for the data/clocks, and /CS1_33, /CS2_33 for the LCD device selects, and /CS_NV for the NVRAM device select. Further, due to I/O limitations, MISO and LOCK functions are shared on the same MCU pin. /CS_NV is used to steer the signals (when the NVRAM is not selected, MISO becomes the LOCK input). The NVRAM utilized here is an "auto-store" device which uses a large value capacitor as a hold-up energy source to allow the memory contents to be save to non-volatile storage when power is lost. This process is transparent to the MCU except that the MCU software must ensure that any data which must be retained is stored to the device as soon as possible to ensure that it will be backed-up on power-loss. *Note: the LCD also uses /BUSY_33, CM/DA_33, and /LCDRST_33 signals.*

The LCD interface is based on the LCD driver ICs that are used on the original IC-900 controller. The "clone" display uses the same protocol to allow seamless connection to either display. This interface is not discussed in detail here, but in general terms, the driver ICs control "segments" on the LCD glass. The segments can be turned on, off, or set to blink. The lcd.c source file manages the segment controls.

SIN and /SOUT_TTL are multi-bit serial data streams that carry control and status information between the RDU and the A-Unit. The SIN data stream is an asynchronous, 31-bit word sent at 4800 baud. SOUT is a synchronous 31-bit word with the same bit-rate. Both protocols are implemented with SSI peripherals on the MCU. This results in the lowest processor overhead, but requires some care at the hardware and driver-software levels to implement. One hardware requirement is that the SOUT signal must be inverted to allow the protocol to be implemented correctly (U5).

UART0 is used for the CLI interface. This is provided as a 3.3V TTL interface at 115200 baud. For the clone interface, these signals exit the RDU module as RS-485 signals which are then converted to RS-232 as part of the A-Unit interface modifications. This interface is multi-tasked to allow the HM-133 multi-function microphone interface to have a command path back to the MCU. This sharing of resources requires an arbitration mechanism. The RTS/CTS control signals from the remote terminal are used to accomplish this in the A-Unit interface modification circuits.

A single ADC input (spare_PD2) is used to capture the output of a light sensing circuit. The voltage captured is in the 0 to +3.3V range and is processed by the system software to apply filtering and level detection.

Two PWM outputs BL_PWM & LED_PWM) are employed that use one of the MCU PWM modules to control the brightness of the various LEDs used in the system. One output drives the back-light LEDs and the other drives the status LEDs. This allows the two to be controlled separately. NAND gates are used to combine the status LED signals (MRX, MTX, and SRX) with the PWM signal.

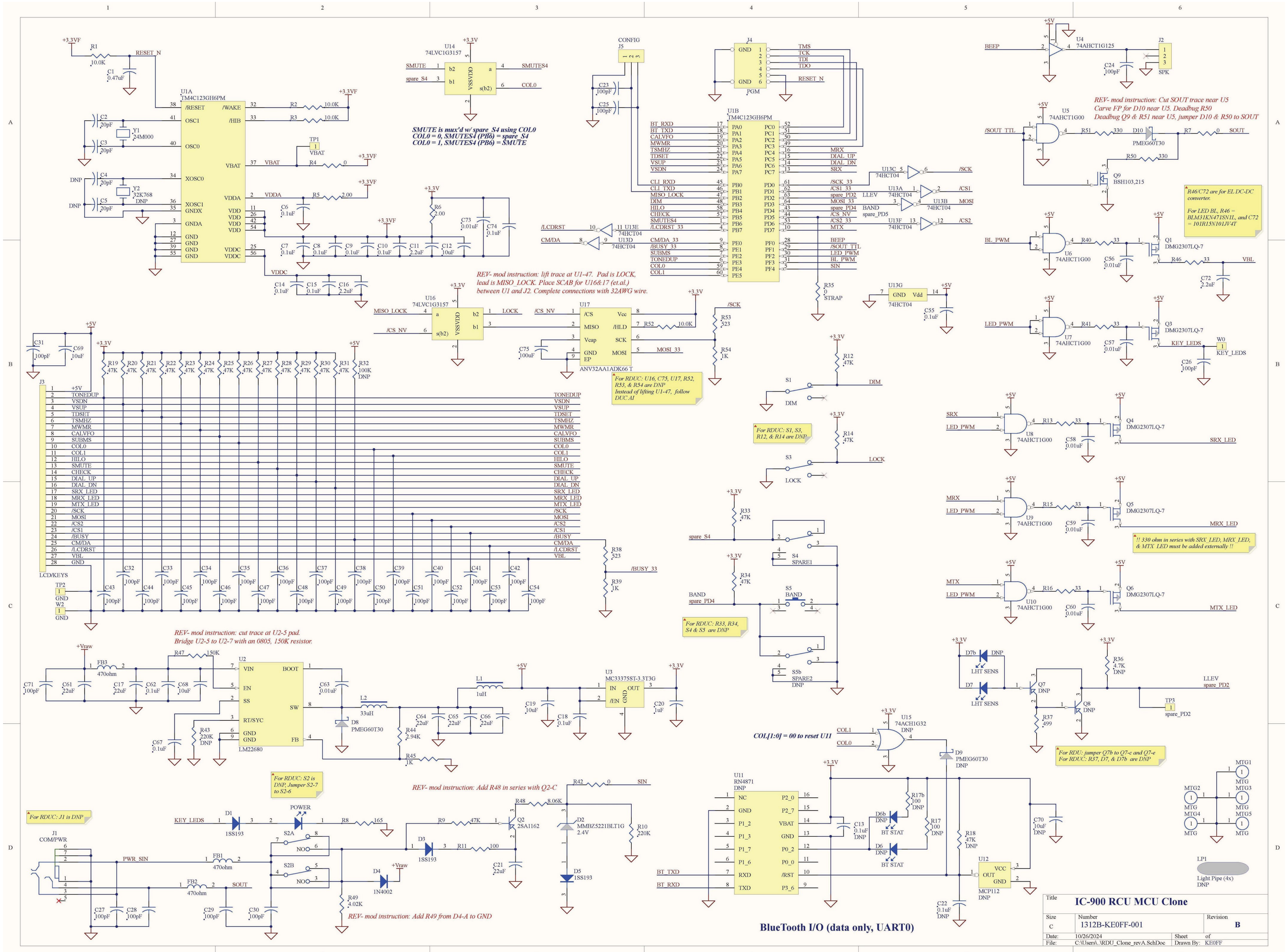
BEEP is a square-wave tone output driven by an MCU timer to produce a single tone signal to drive a piezo-electric speaker.

Power Supplies ('n stuff)

The RDU uses a buck-mode switching regulator to produce 5Vdc power from the approximately 11V(ave) power/SIN signal. An LM22680 IC is used with a 33uH inductor. An SOT-223 package 3.3V regulator produces power for the MCU using the 5Vdc as its source.

The data separator circuit (Q2) is derived from the ICOM design and removes the SIN data signal from the multiplexed power signal. R49 is required to compensate for the pulsed current behavior of the LM22680 (the original ICOM design used an LM7805 linear device).

J3 is a "ribbon" cable footprint that matches the cable used on the original ICOM design. This allows the original LCD/switch board to be used with the RDUC circuit board.



REV- mod instruction: Cut SOUT trace near U5
Carve FP for D10 near U5. Deadbug R50
Deadbug Q9 & R51 near U5, jumper D10 & R50 to SOUT
SMUTE is mux'd w/ spare_S4 using COL0
COL0 = 0, SMUTES4 (PB6) = spare_S4
COL0 = 1, SMUTES4 (PB6) = SMUTE
REV- mod instruction: lift trace at U1-47. Pad is LOCK, lead is MISO_LOCK. Place SCAB for U16&17 (et.al.) between U1 and J2. Complete connections with 32AWG wire.
For RDUC: U16, C75, U17, R52, R53, & R54 are DNP Instead of lifting U1-47, follow DUC AI
R46/C72 are for EL DC-DC converter.
For LED BL, R46 = BLM31KN471SN1L, and C72 = 101R15N101JV4T
For RDUC: S1, S3, R12, & R14 are DNP
For RDUC: R33, R34, S4 & S5 are DNP
!! 330 ohm in series with SRX_LED, MRX_LED, & MTX_LED must be added externally !!
REV- mod instruction: cut trace at U2-5 pad. Bridge U2-5 to U2-7 with an 0805, 150K resistor.
For RDUC: S2 is DNP, Jumper S2-7 to S2-6
REV- mod instruction: Add R48 in series with Q2-C
REV- mod instruction: Add R49 from D4-A to GND
For RDUC: J1 is DNP
COL[1:0] = 00 to reset U11
For RDU: jumper Q7b to Q7-c and Q7-e
For RDUC: R37, D7, & D7b are DNP
BlueTooth I/O (data only, UART0)
Title IC-900 RCU MCU Clone
Size C
Number 1312B-KE0FF-001
Revision B
Date: 10/26/2024
Sheet of
File: C:\Users\.\RDU_Clone_revA.SchDoc
Drawn By: KE0FF

DU Clone (ICOM RDU LCD Emulator)

The DU Clone (DUC) uses a Tiva TM4C123GHPM MCU to receive SSI control signals from the RDU controller. These signals (MISO_LOCK, MOSI_33, /SCK_33, /CS1_33, /CS2_33, /BUSY_33, CM/DA_33, and /LCDRST_33) are decoded and interpreted with an aim at maintaining symbology similar to the ICOM LCD glass on a medium-scale dot-matrix LCD display. The conversion is accomplished in SW and is transparent to the RDU controller.

MCU

The MCU circuit is the same used for the RDUC. The power required for the DUC is provided by the RDUC so there are no power supplies in the DUC design other than a 3.3V LDO (U6). The DUC MCU uses a single SSI module and some GPIO pins for the RDUC SSI interface, and several GPIO pins to form the parallel interface with the dot-matrix LCD module which uses HCT and LXC devices to accomplish the 3.3V<-> 5V logic conversion (U2 & U5). U3 combines the two chip selects into a single signal for the SSI logic. The two signals are also captured by GPIO pins to help decode the input data.

NVRAM S.C.A.B.

The NVRAM feature of the RDUC was a post-layout change that required a Supplemental Circuit Addition Board (SCAB). This circuit was included in the DUC design (U16a and U17a) to alleviate the need for the SCAB on the RDUC.

Switch Matrix and Backlight LEDs

The original ICOM design for the switch matrix is replicated on the DUC. In addition, numerous LEDs are placed to accomplish button backlighting. These LEDs are driven from the RDUC LED_PWM source.

The status LEDs, MRX, MTX, & SRX are also placed on the DUC.

The dial is implemented as a separate daughter board to accommodate the mechanical aspects of this encoder. It features its own backlight LEDs which are powered by the LED backlight supply.

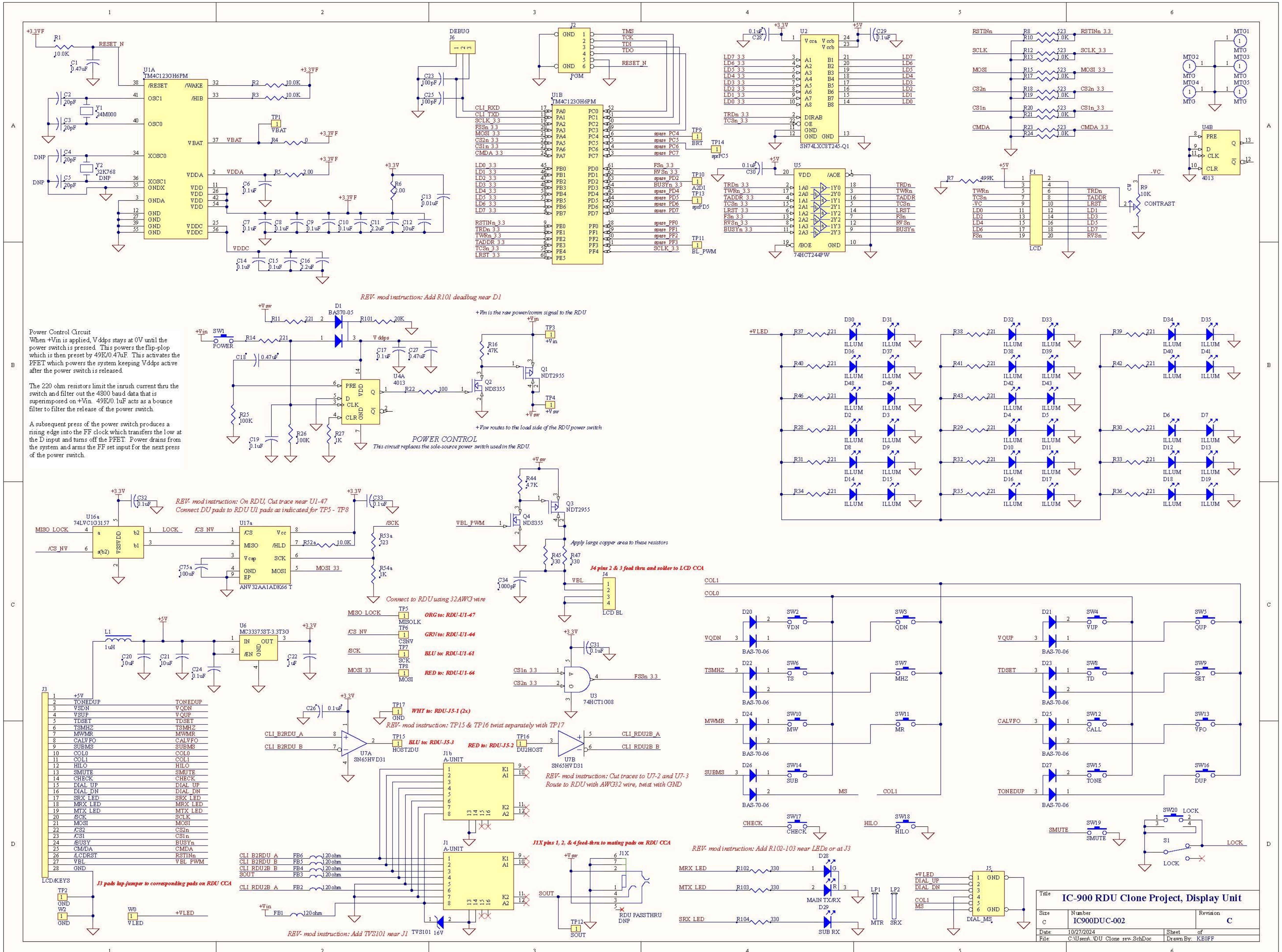
The LCD backlight requires some buffering to accommodate the relatively large current required. Q4 and Q3 provide this circuitry.

To simplify procurement, the ON/OFF type power switch used in the original ICOM design was not used here. Instead, the same momentary switch used for the user buttons was connected to a flip-flop circuit (U4a) to produce “push-on/push-off” power control. The momentary switch supplies power to the circuit and initializes the flip-flop into the power-on state. This ensures that the circuit is powered and the RDUC/DUC system power is on. A subsequent press of the power button causes the flip-flop to advance to the next state (off). Q1 is the main power switch element for the system.

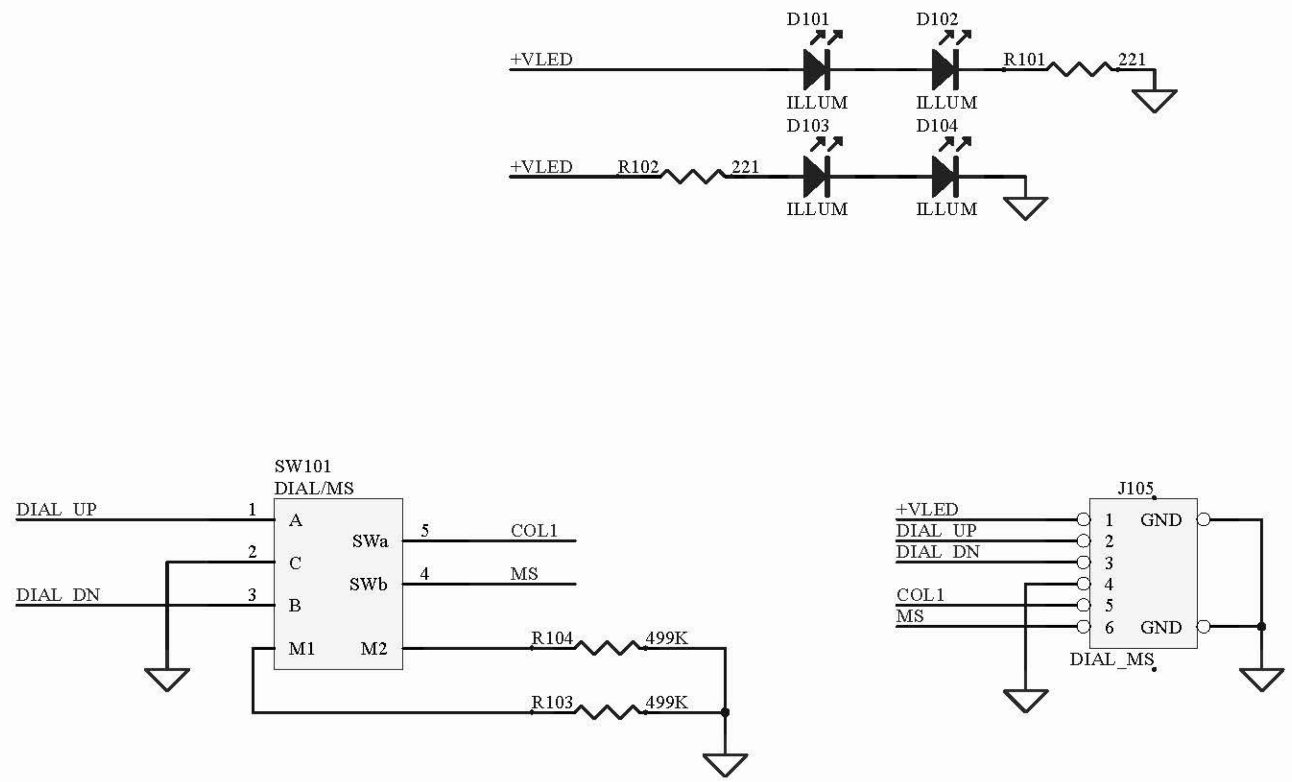
RS-485 Transcievers

The RDUC serial port is buffered with an SN65HVD31 (U7) to produce a full-duplex, RS-485 data link.

An RJ-45 connector using standard Ethernet pairing is used to convey the RS-485 data along with the SIN/Power and SOUT signals (each on its own pair).



Power Control Circuit
When +Vin is applied, Vddps stays at 0V until the power switch is pressed. This powers the flip-plop which is then preset by 49K/0.47uF. This activates the PFET which powers the system keeping Vddps active after the power switch is released.
The 220 ohm resistors limit the inrush current thru the switch and filter out the 4800 baud data that is superimposed on +Vin. 49K/0.1uF acts as a bounce filter to filter the release of the power switch.
A subsequent press of the power switch produces a rising edge into the FF clock which transfers the low at the D input and turns off the PFET. Power drains from the system and arms the FF set input for the next press of the power switch.
POWER CONTROL
This circuit replaces the sole-source power switch used in the RDU.
REV- mod instruction: Add R101 deadbug near D1
+Vin is the raw power/comm signal to the RDU
+Vsw routes to the load side of the RDU power switch
REV- mod instruction: On RDU, Cut trace near U1-47
Connect DU pads to RDU U1 pads as indicated for TP5 - TP8
Connect to RDU using 32AWG wire
ORG to: RDU-U1-47
GRN to: RDU-U1-44
BLU to: RDU-U1-61
RED to: RDU-U1-64
Apply large copper area to these resistors
J4 pins 2 & 3 feed thru and solder to LCD CCA
WHT to: RDU-J5-1 (2x)
REV- mod instruction: TP15 & TP16 twist separately with TP17
BLU to: RDU-J5-3
RED to: RDU-J5-2
REV- mod instruction: Cut traces to U7-2 and U7-3
Route to RDU with AWG32 wire, twist with GND
J1X pins 1, 2, & 4 feed-thru to mating pads on RDU CCA
REV- mod instruction: Add R102-103 near LEDs or at J3
J3 pads lap-jumper to corresponding pads on RDU CCA
REV- mod instruction: Add TVS101 near J1
U1A TM4C123GH6PM
U1B TM4C123GH6PM
U2 SN74LXCST245-Q1
U5 74HCT244PW
U4B 4013
U4A 4013
U16a 74LVC1G3157
U17a ANV32AA1ADK66 T
U6 MC33375ST-3.3T3G
U3 74HCT1G08
U7A SN65HVD31
U7B SN65HVD31
J3 LCD/KEYS
J1 A-UNIT
J1b A-UNIT
J1X RDU PASSTHRU DNP
J5 DIAL_MS
J4 LCD BL
J2 FGM
J6 DEBUG
P1 LCD
Title IC-900 RDU Clone Project, Display Unit
Size C
Number IC900DUC-002
Revision C
Date: 10/27/2024
Sheet of
File: C:\Users\...\DU_Clone_rev.SchDoc
Drawn By: KE0FF



Title	IC-900F DU Dial Board		
Size A	Number		Revision -
Date:	10/27/2024	Sheet of	
File:	C:\Users\..\DIAL.SchDoc	Drawn By:	

A-Unit Modifications

In order to get the benefit of the HM-133 microphone features, the new controller requires that the A-unit be modified to facilitate the serial connection from the HM-133 to the controller. Figure 1 illustrates a modified A-unit for reference. The hardware to accomplish this interface also provides an RJ-45 connection for the RDUC. A standard Ethernet cable is used. ***Note: The RDU RJ-45 connection is NOT compatible with the HM-133. Care must be exercised to avoid connecting the HM-133 to the RDU connector***.

Figure 1: Modified IC-900 A-Unit and HM-133 Adapter (the controller connection unit is pictured in the lower right of the image)

The A-Unit modification features two circuit boards and utilizes an ATtiny-816 MCU to manage the microphone and RDUC serial port connections. The MCU uses UART, GPIO, and ADC peripherals to accomplish the tasks needed:

1) Detect if an HM-133 or HM-14 microphone is connected and set the UART switch appropriately

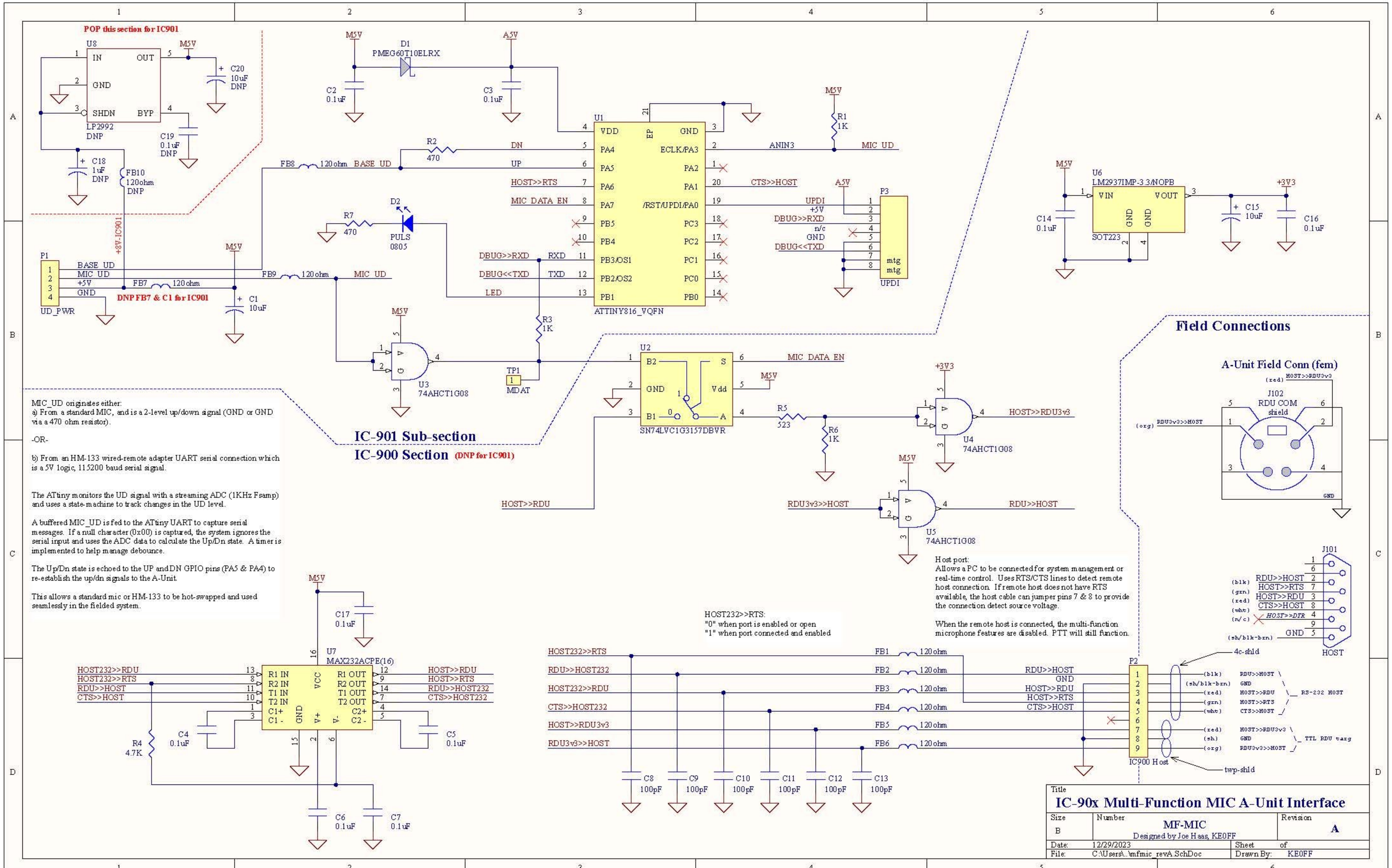
2) Monitor RTS from the remote terminal and set the UART switch and CTS signal appropriately (allows the remote terminal to access the RDUC UART with minimal interference to the HM-133 operation)

3) Convert RS-485 signaling to RS-232 for the remote terminal

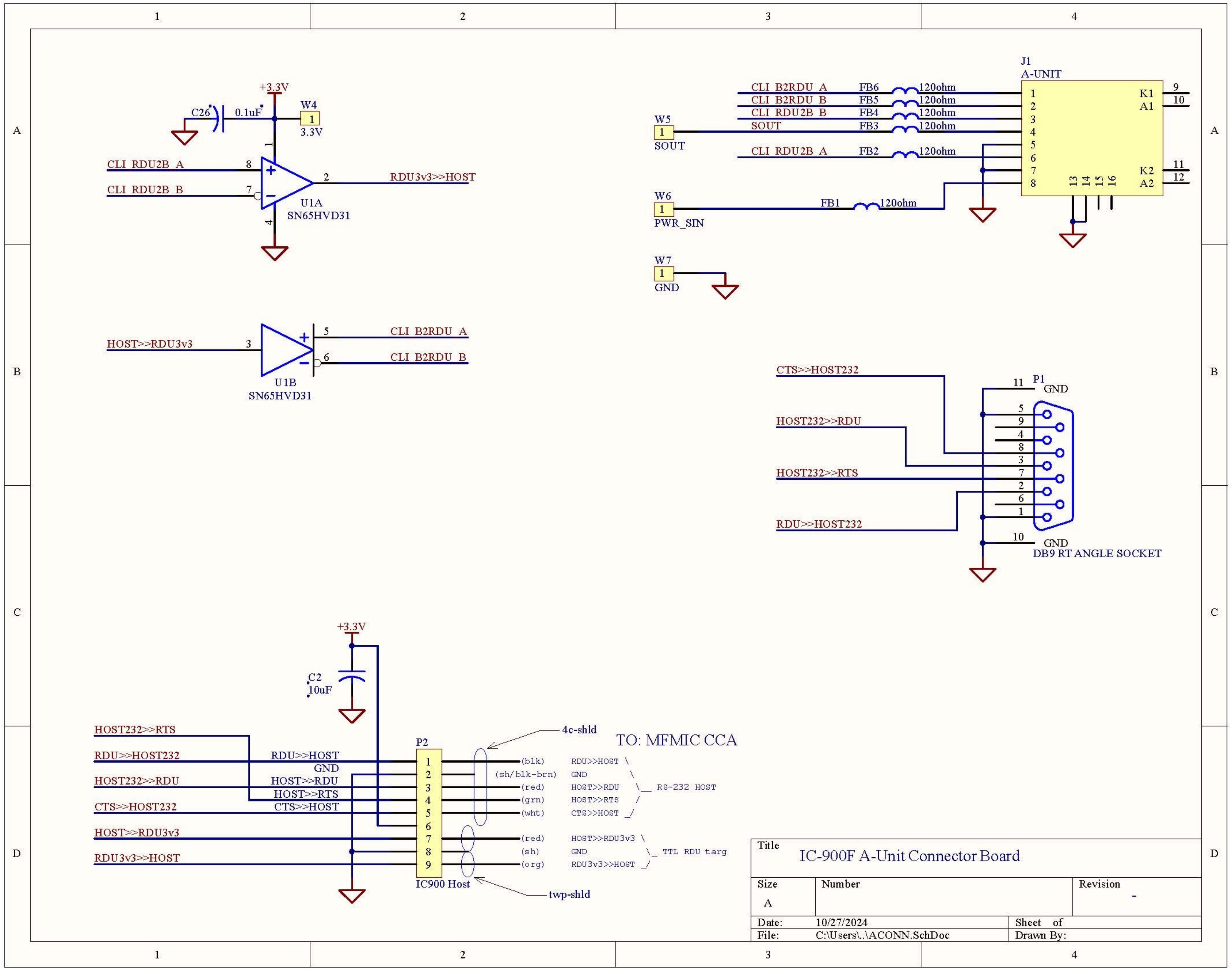
The HM-133 uses the U/D signal to send serial data (at 115200 baud) to the RDUC allowing the RDUC to accept keypress information from the HM-133 multi-function microphone. The MCU monitors the analog state of the U/D signal as well as any serial data present. If serial data is detected, the MCU assumes that an HM-133 is connected and it ignores the analog U/D status. If the analog U/D signal produces signals that match the HM-14 up/down signal, the MCU “repeats” the analog U/D status using GPIO pins and a 470 ohm resistor.

The MCU also monitors the RTS signal and activates the serial switch (U2) to route the host serial data to the RDUC. Otherwise, the serial path is routed from the HM-133 to the RDUC. In addition, the MCU activates or deactivates CTS to reflect the U2 switch setting.
D2 is a status LED that is used by the MCU software to indicate “I’m alive” status.

The second board contains the RS-485 transceiver and the DB9 & RJ-45 connectors. A set of 3D-printed parts are used to act as an enclosure for the two circuit boards. Another set of 3D-printed parts secures the add-on cable inside the A-Unit enclosure. The result is reasonably asthetic and allows the A-Unit to be used with legacy controllers without undoing the modifications.



POP this section for IC901
U8
LP2992
DNP
D1
PMEG60T10ELRX
U1
ATTINY816_VQFN
U6
LM2937IMP-3.3/NOPB
SOT223
P3
UPDI
P1
UD_PWR
DNP FB7 & C1 for IC901
+8V-IC901
U3
74AHCT1G08
TP1
MDAT
U2
SN74LVC1G3157DBVR
U4
74AHCT1G08
U5
74AHCT1G08
Field Connections
A-Unit Field Conn (fem)
J102
RDU COM
shield
J101
HOST
IC-901 Sub-section
IC-900 Section
(DNP for IC901)
MIC_UD originates either:
a) From a standard MIC, and is a 2-level up/down signal (GND or GND via a 470 ohm resistor).
-OR-
b) From an HM-133 wired-remote adapter UART serial connection which is a 5V logic, 115200 baud serial signal.
The ATtiny monitors the UD signal with a streaming ADC (1KHz Fsamp) and uses a state-machine to track changes in the UD level.
A buffered MIC_UD is fed to the ATtiny UART to capture serial messages. If a null character (0x00) is captured, the system ignores the serial input and uses the ADC data to calculate the Up/Dn state. A timer is implemented to help manage debounce.
The Up/Dn state is echoed to the UP and DN GPIO pins (PA5 & PA4) to re-establish the up/dn signals to the A-Unit.
This allows a standard mic or HM-133 to be hot-swapped and used seamlessly in the fielded system.
U7
MAX232ACPE(16)
HOST232>>RTS:
"0" when port is enabled or open
"1" when port connected and enabled
Host port:
Allows a PC to be connected for system management or real-time control. Uses RTS/CTS lines to detect remote host connection. If remote host does not have RTS available, the host cable can jumper pins 7 & 8 to provide the connection detect source voltage.
When the remote host is connected, the multi-function microphone features are disabled. PTT will still function.
P2
IC900 Host
RS-232 HOST
TTL RDU targ
Title
IC-90x Multi-Function MIC A-Unit Interface
Size
B
Number
MF-MIC
Designed by Joe Haas, KE0FF
Revision
A
Date:
12/29/2023
Sheet
of
File:
C:\Users\..\mfmic_revA.SchDoc
Drawn By:
KE0FF

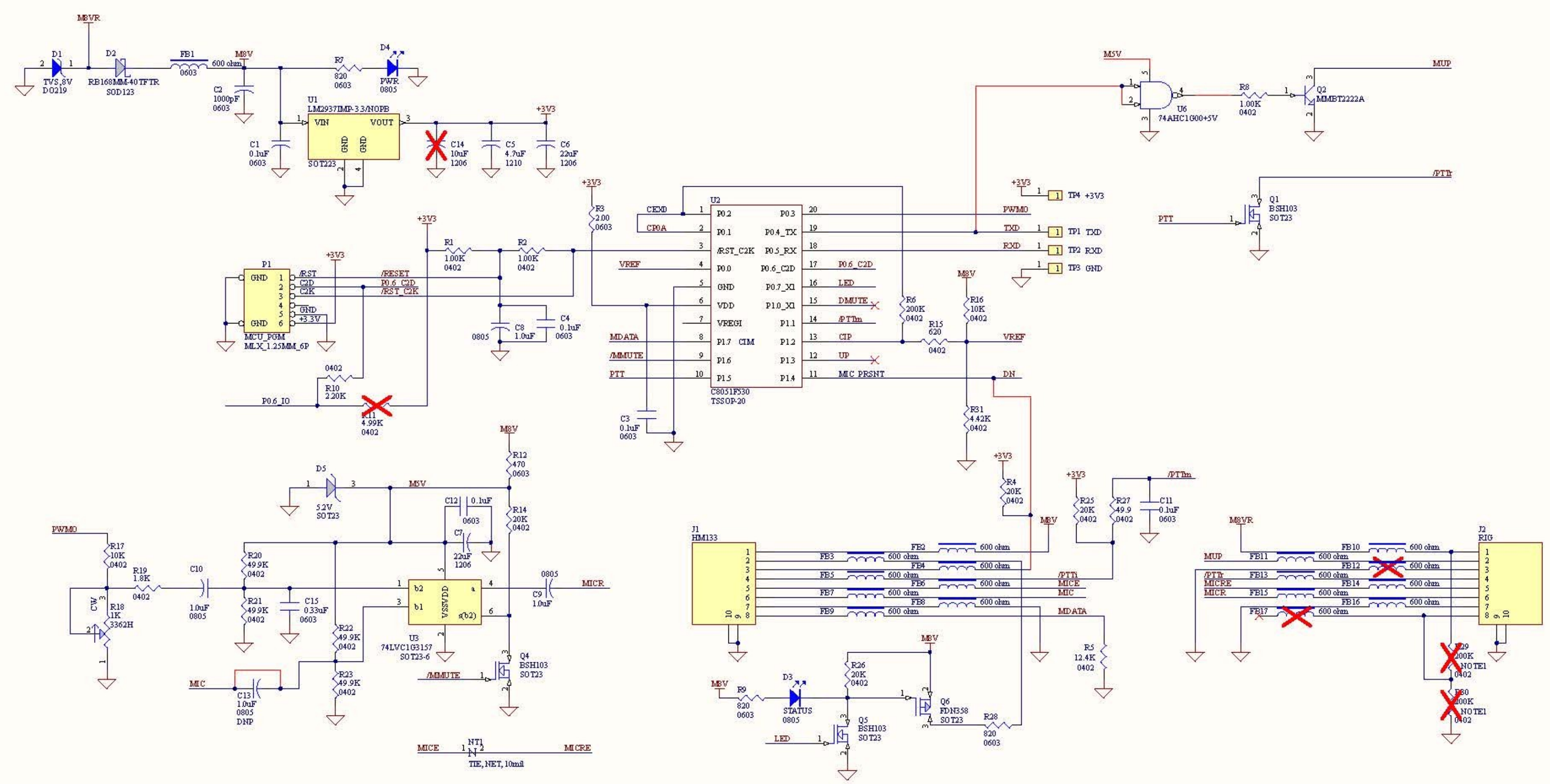

+3.3V
C26
0.1uF
W4
3.3V
CLI_RDU2B_A
CLI_RDU2B_B
U1A
SN65HVD31
RDU3v3>>HOST
HOST>>RDU3v3
U1B
SN65HVD31
CLI_B2RDU_A
CLI_B2RDU_B
J1
A-UNIT
CLI_B2RDU_A FB6 120ohm
CLI_B2RDU_B FB5 120ohm
CLI_RDU2B_B FB4 120ohm
SOUT FB3 120ohm
CLI_RDU2B_A FB2 120ohm
W5
SOUT
W6
PWR_SIN
FB1 120ohm
W7
GND
K1
A1
K2
A2
CTS>>HOST232
HOST232>>RDU
HOST232>>RTS
RDU>>HOST232
P1
GND
DB9 RT ANGLE SOCKET
+3.3V
C2
10uF
HOST232>>RTS
RDU>>HOST232
HOST232>>RDU
CTS>>HOST232
HOST>>RDU3v3
RDU3v3>>HOST
RDU>>HOST
GND
HOST>>RDU
HOST>>RTS
CTS>>HOST
P2
IC900 Host
4c-shld
twp-shld
TO: MFMIC CCA
(blk) RDU>>HOST \
(sh/blk-brn) GND \
(red) HOST>>RDU __ RS-232 HOST
(grn) HOST>>RTS /
(wht) CTS>>HOST _/
(red) HOST>>RDU3v3 \
(sh) GND _ TTL RDU targ
(org) RDU3v3>>HOST _/
Title
IC-900F A-Unit Connector Board
Size
A
Number
Revision
-
Date: 10/27/2024
Sheet of
File: C:\Users\..\ACONN.SchDoc
Drawn By:

HM-133 Adapter

The HM-133 adapter allows an HM-133 to connect to any radio that accepts a DTMF microphone. It was originally developed solely for this purpose but was later modified to allow all of the HM-133 button codes to be serially transmitted to a host controller allowing for multiple functions to be accessed (in addition to producing DTMF tones when transmitting).

The adapter features a Silicon Labs C8051-C531 MCU which captures the HM-133 serial messages and converts them to standard UART messages using a custom protocol. The MCU also produces DTMF tones and combines them with the microphone circuit to allow the system to provide DTMF capability.

RJ-45 connectors are provided for the HM-133 connection and the radio connection. A short adapter cable is generally needed to connect the HM-133 Adapter to the radio of choice.



Title	HM133 DTMF Adapter, Wired-Remote	
Size: C	Number: HM133_DTMF_sch	Revision: 001C
Date: 10/27/2024	Sheet of	
File: C:\Users\.\HM133_DTMF.SchDoc	Drawn By:	