POWER TELEMETRY FOR REMOTE PAs

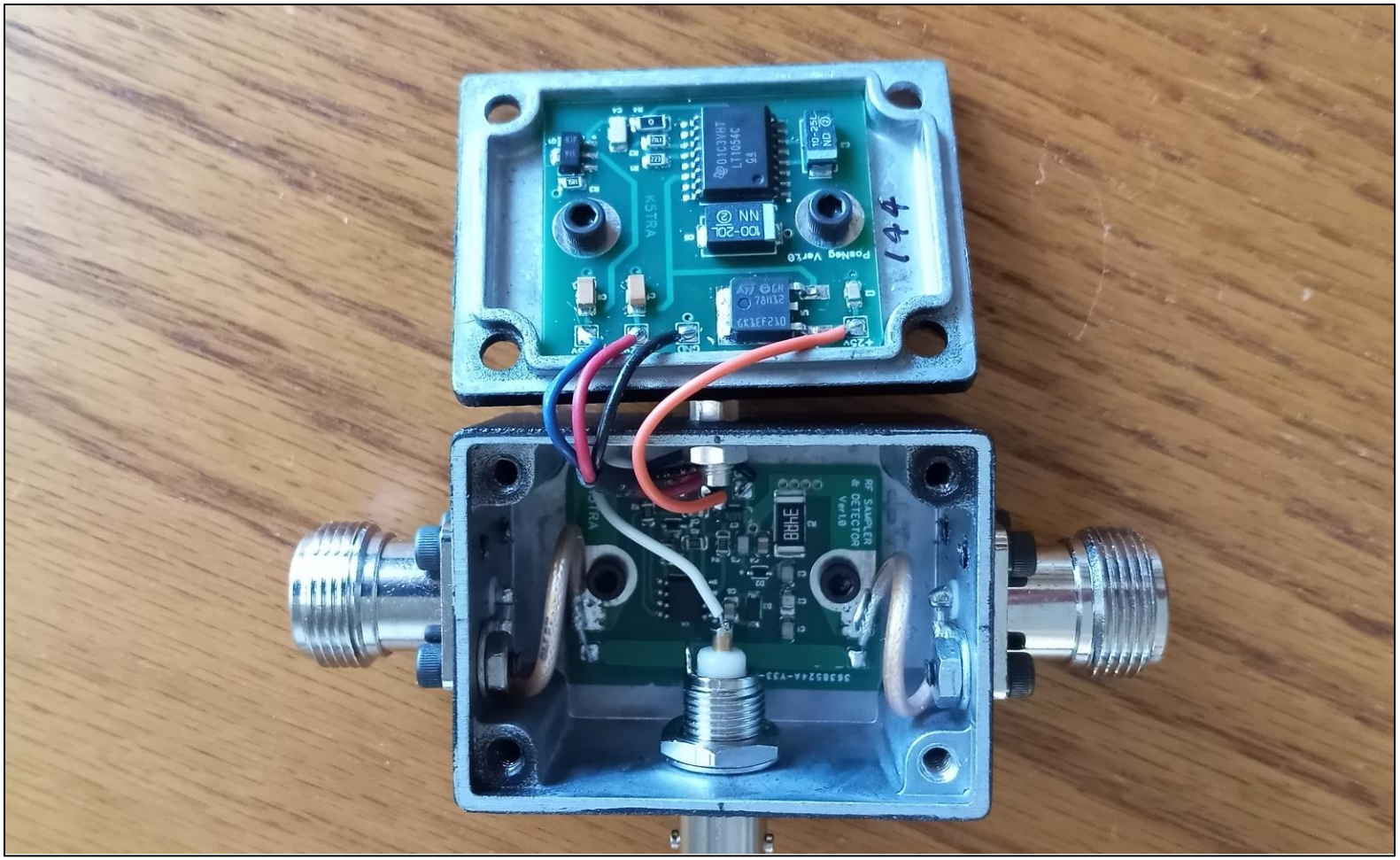
DESIGNS and LAYOUTS for remote sensing
144, 222, 432, 902 & 1296 MHz

OVERVIEW

- Remote PA power level sensing needed
- No possibility of additional lines from attic in conduit
- Ethernet LAN solution
- Band specific sensor-detector for each band (5 total)
- Detectors are temp compensated and knee linearized
- Arduinos provide AD and DA conversions
- LP-500 display provides power display in radio shack
- Four PC boards with surface mount assembly
- Five detector outputs are summed and digitized
- In shack, analog voltage is recovered and passed to LP-500 display

RF SAMPLER-DETECTORS

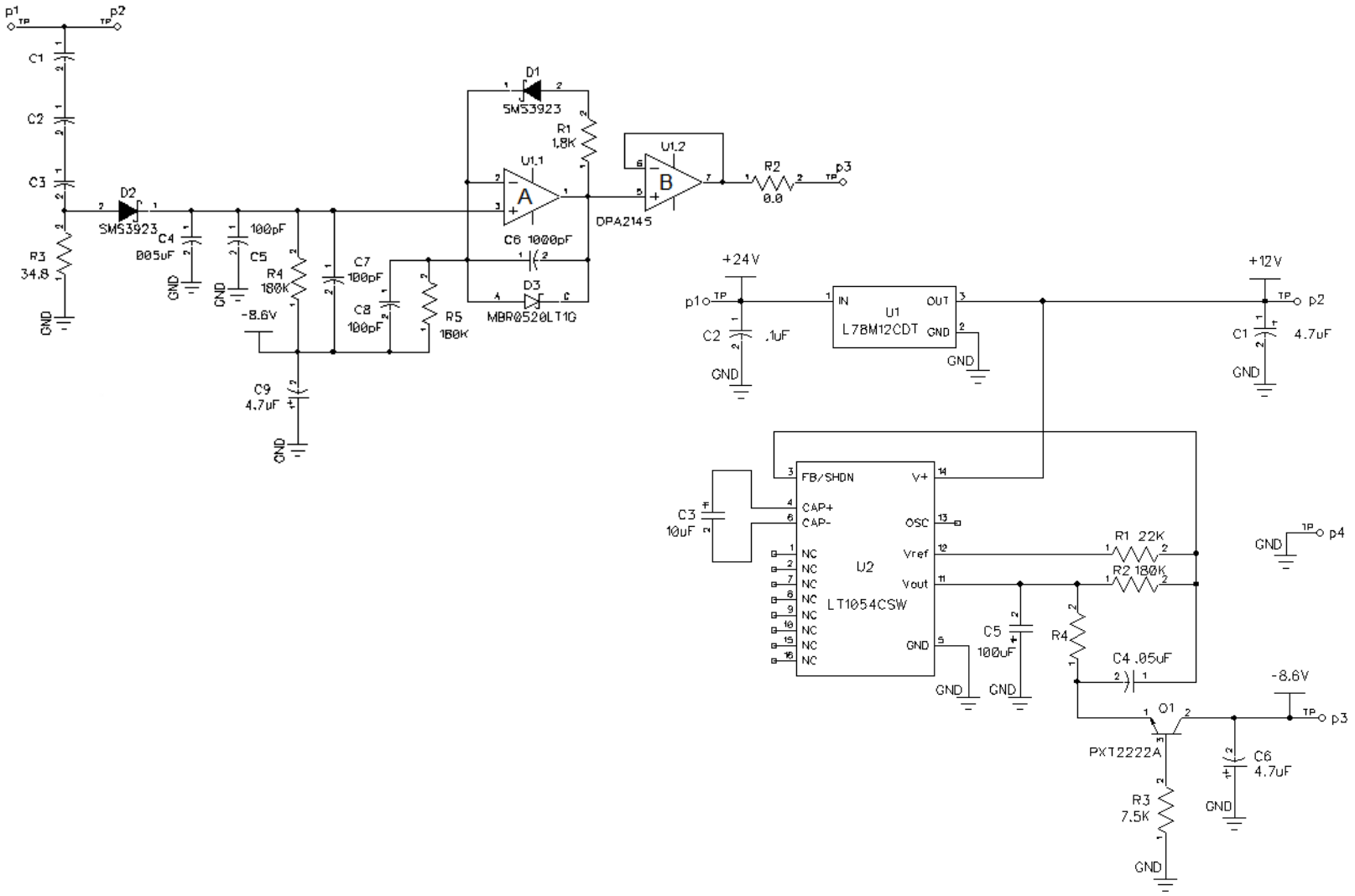
RF SAMPLER-DETECTORS



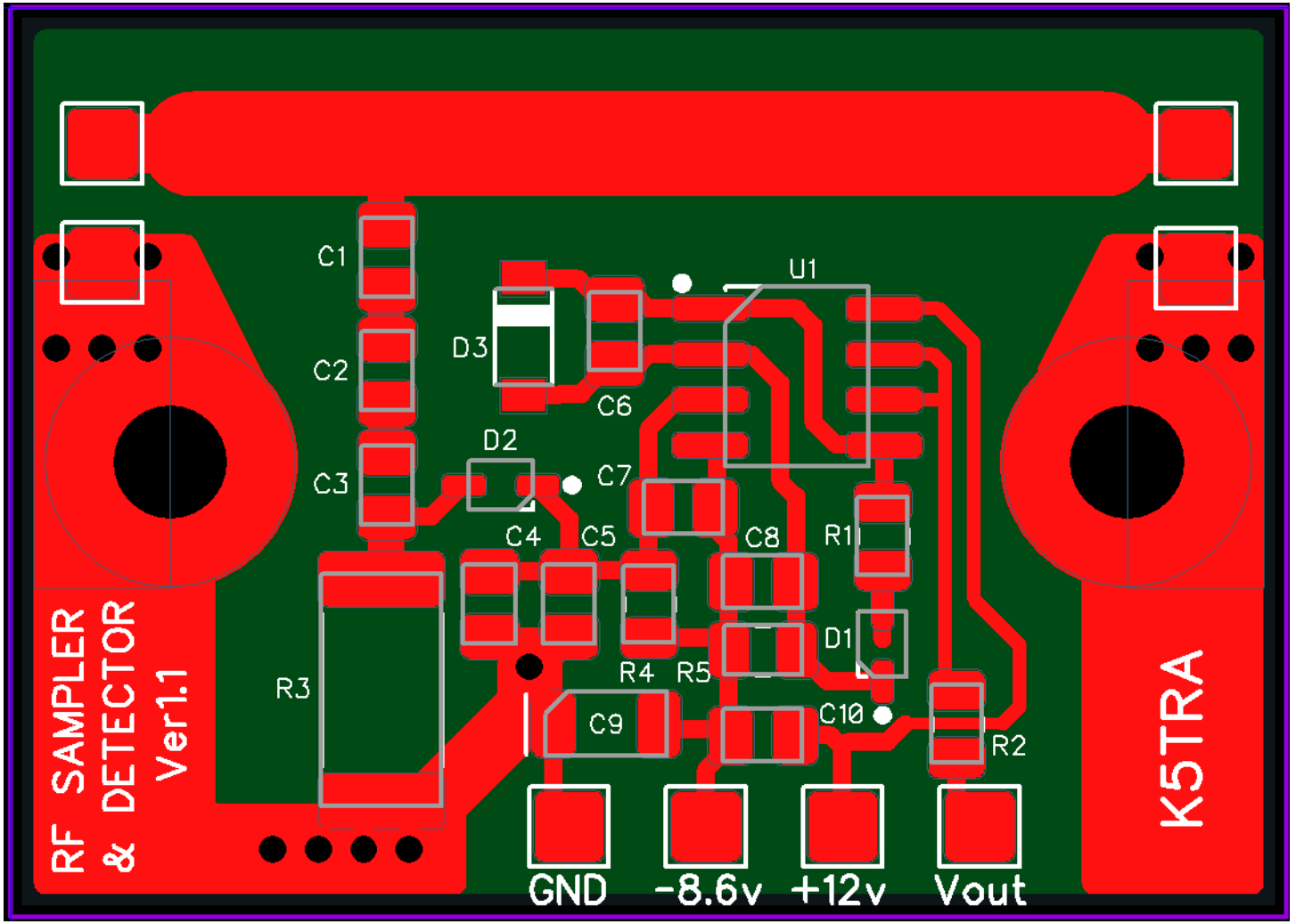
RF SAMPLER-DETECTOR CIRCUIT SUMMARY

- -27.5 dB coupling level
- RC voltage sampler
- OP amp temp and diode corner linearization
- Bias board with negative voltage source
- Detectors are powered from PA +25v source

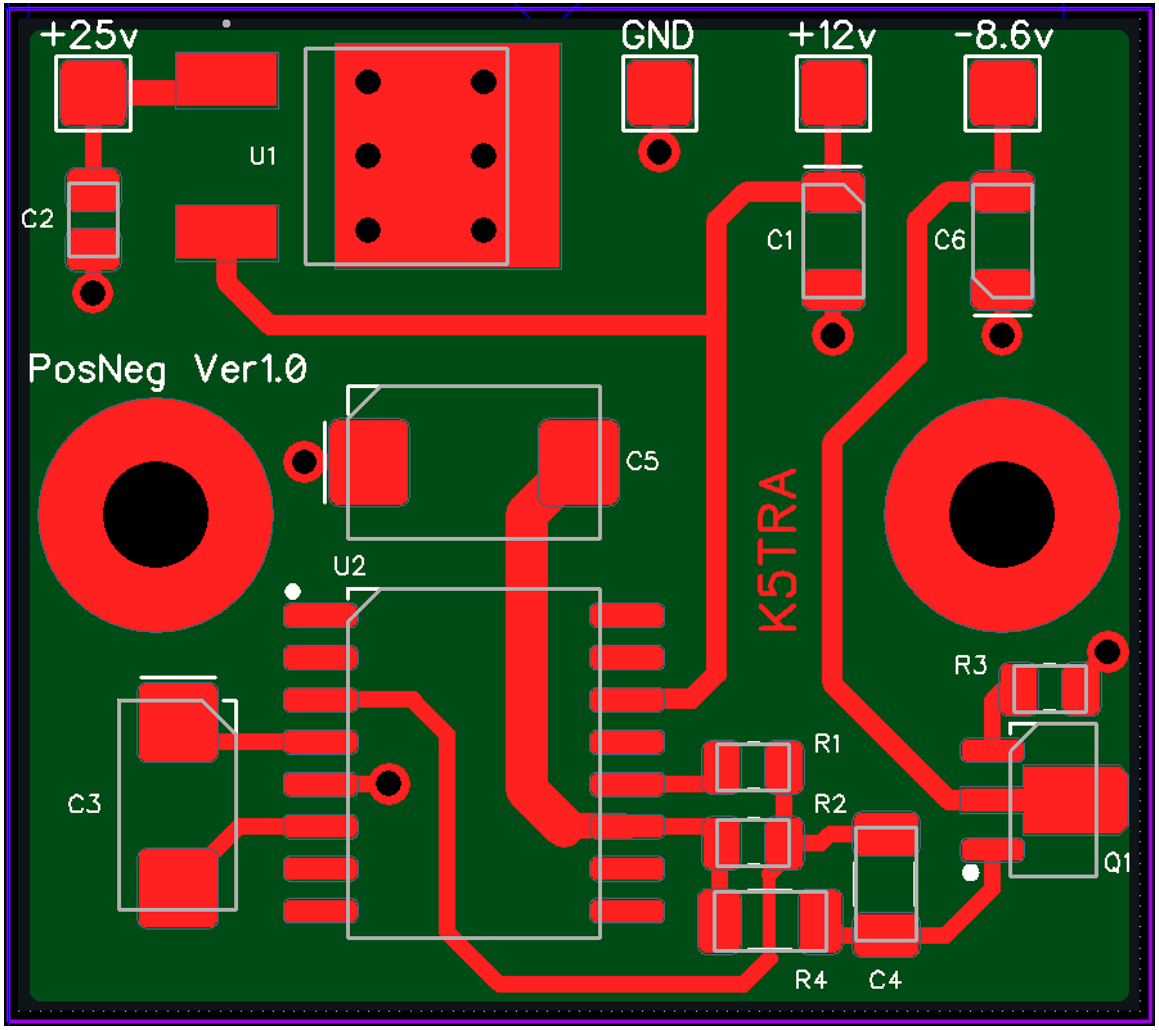
RF SAMPLER-DETECTOR SCHEMATIC



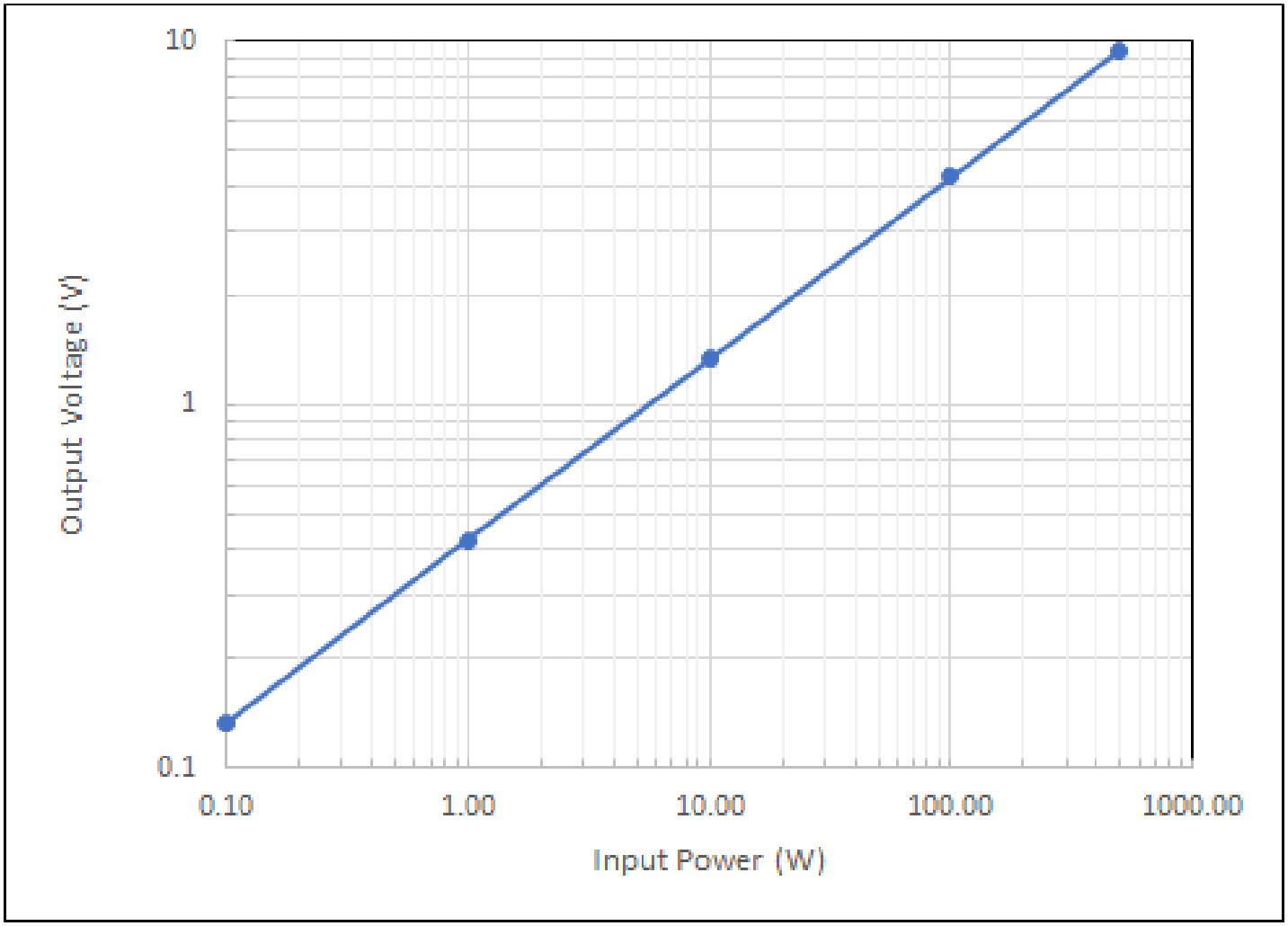
RF SAMPLER-DETECTOR BOARD



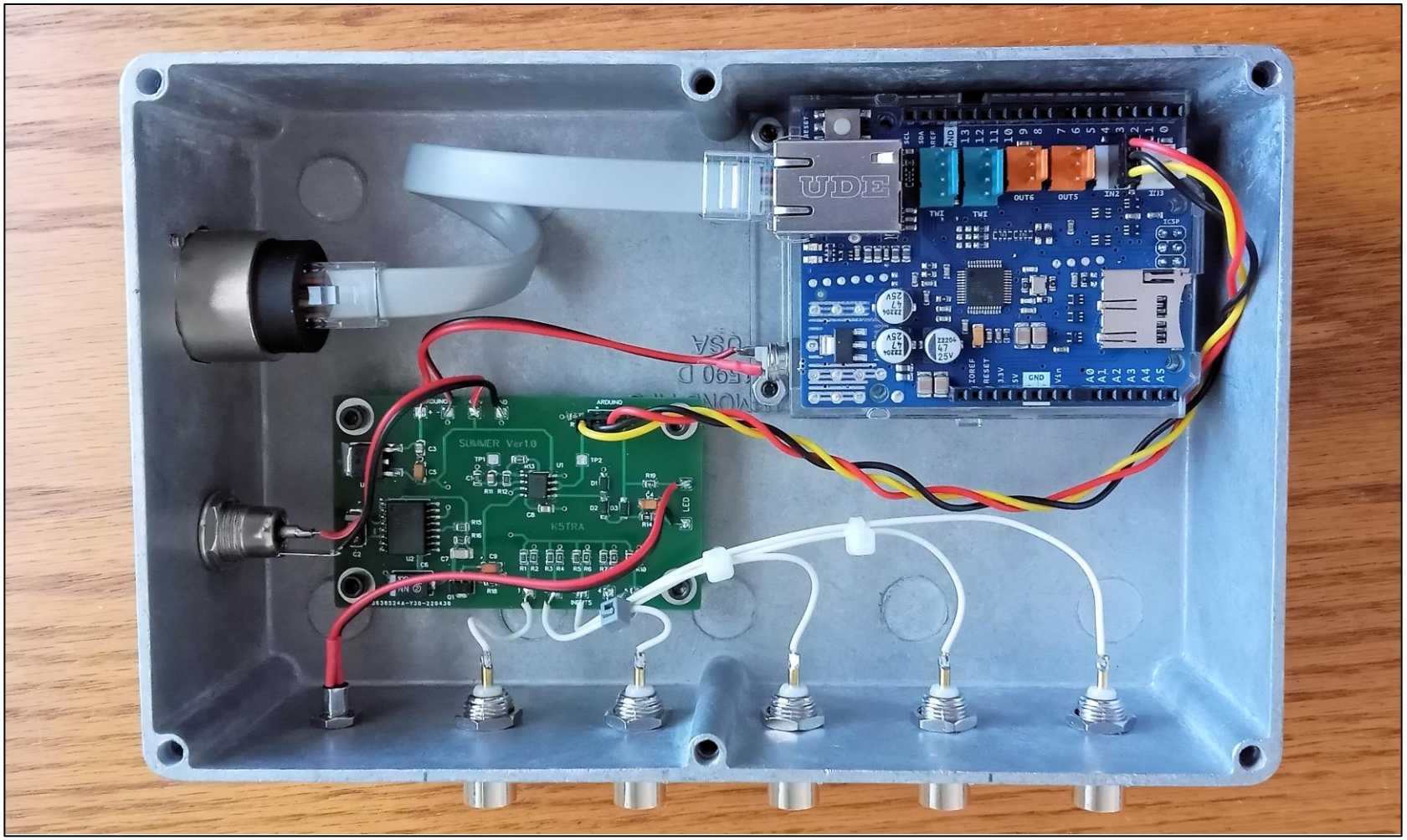
DETECTOR BIAS BOARD



SAMPLER-DETECTOR RESPONSE



REMOTE SUMMER & ARDUINO

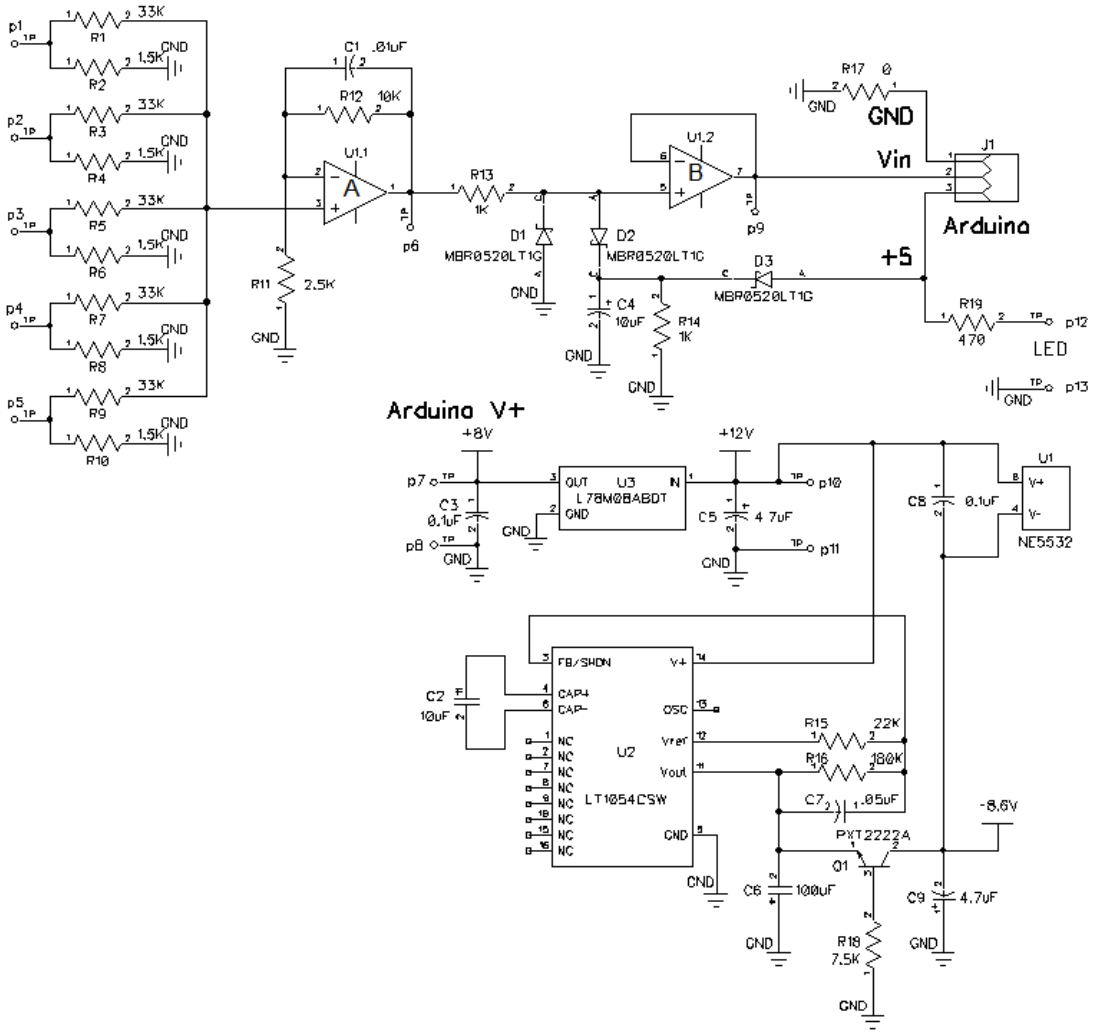


REMOTE SUMMER & ARDUINO

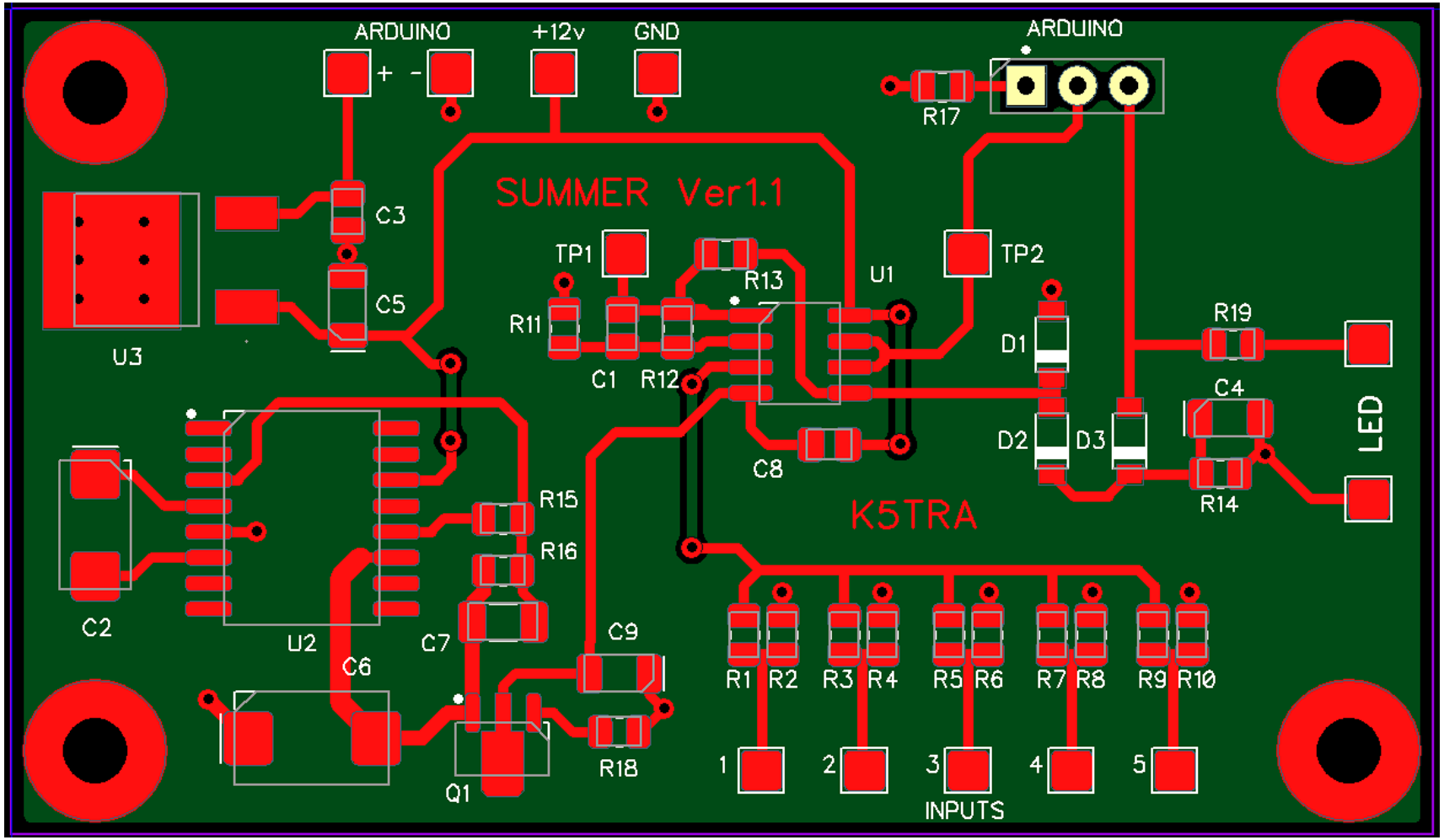
5-WAY SUMMER CIRCUIT SUMMARY

- Unity gain voltage summer and unity gain buffer
- Max voltage clamp at +5v for Arduino input
- Bias circuit with negative voltage source
- Bias circuit provides +8v for Arduino
- Arduino provides AD conversion and TCP/IP LAN communication

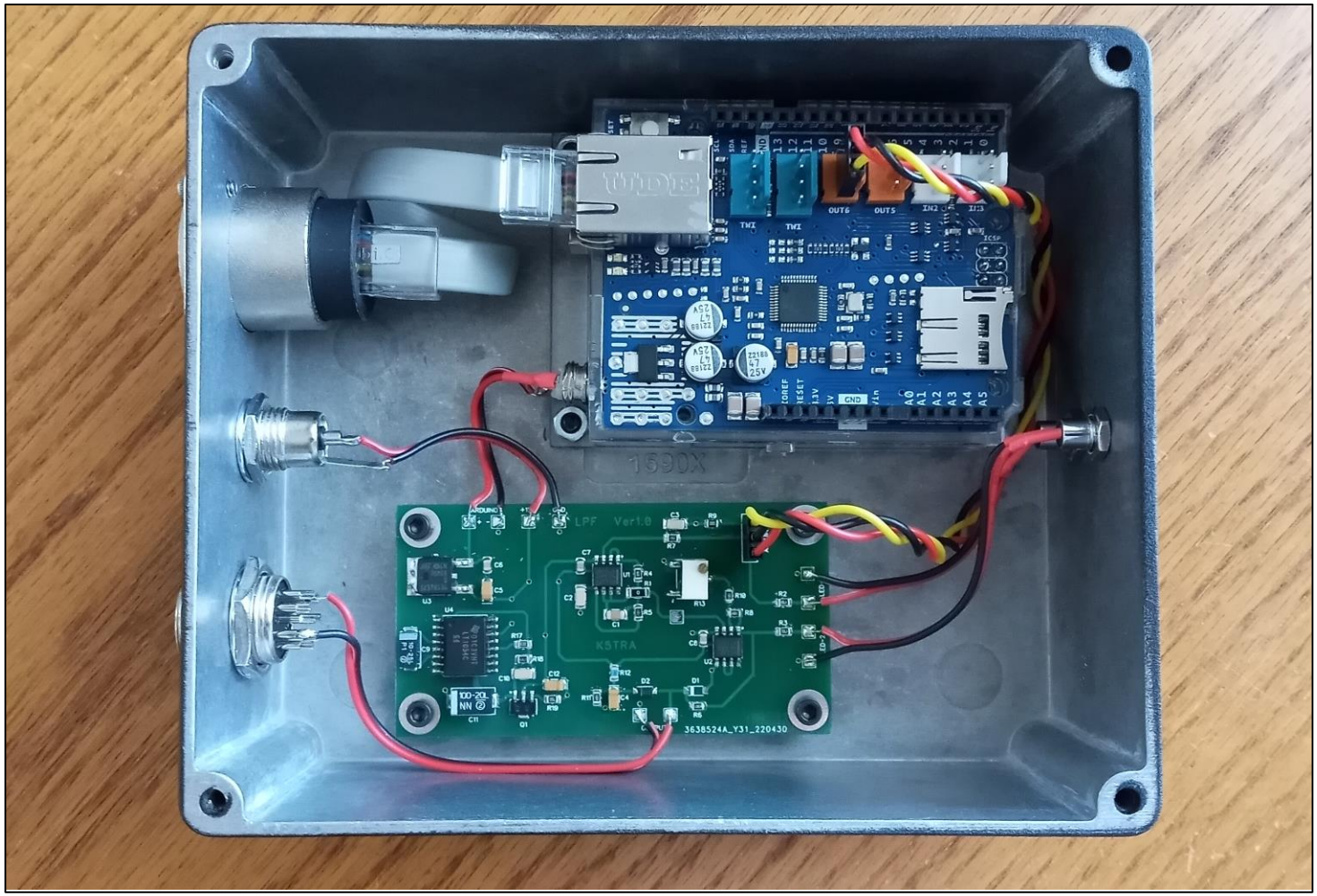
5-WAY SUMMER SCHEMATIC



5-WAY SUMMER BOARD

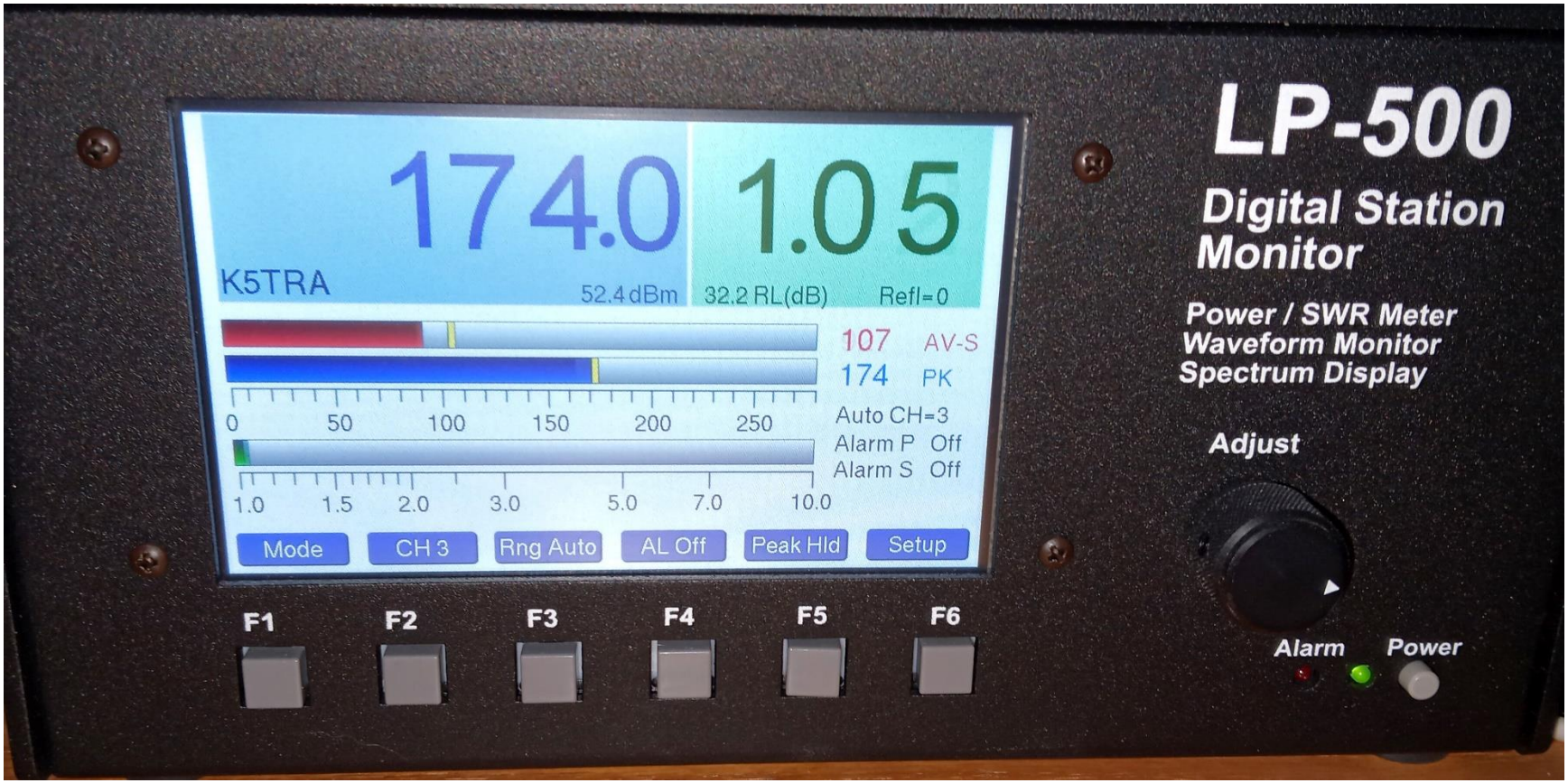


IN SHACK TELEMETRY RECEIVER



IN SHACK TELEMETRY RECEIVER

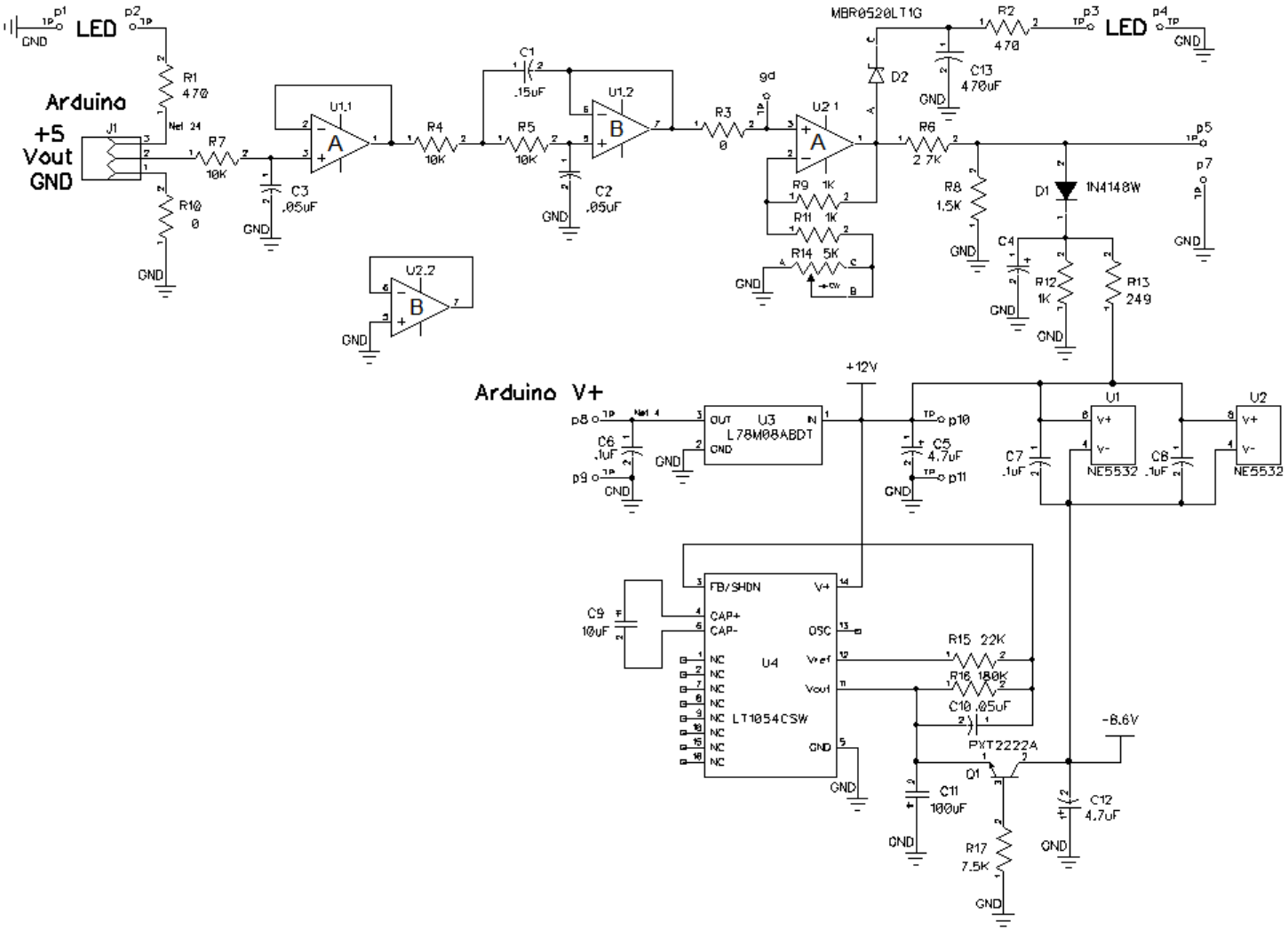
DISPLAY


174.0
1.05
K5TRA
52.4 dBm
32.2 RL(dB)
Refl=0
107 AV-S
174 PK
0 50 100 150 200 250
Auto CH=3
Alarm P Off
Alarm S Off
1.0 1.5 2.0 3.0 5.0 7.0 10.0
Mode
CH 3
Rng Auto
AL Off
Peak Hld
Setup
F1 F2 F3 F4 F5 F6
LP-500
Digital Station Monitor
Power / SWR Meter
Waveform Monitor
Spectrum Display
Adjust
Alarm
Power

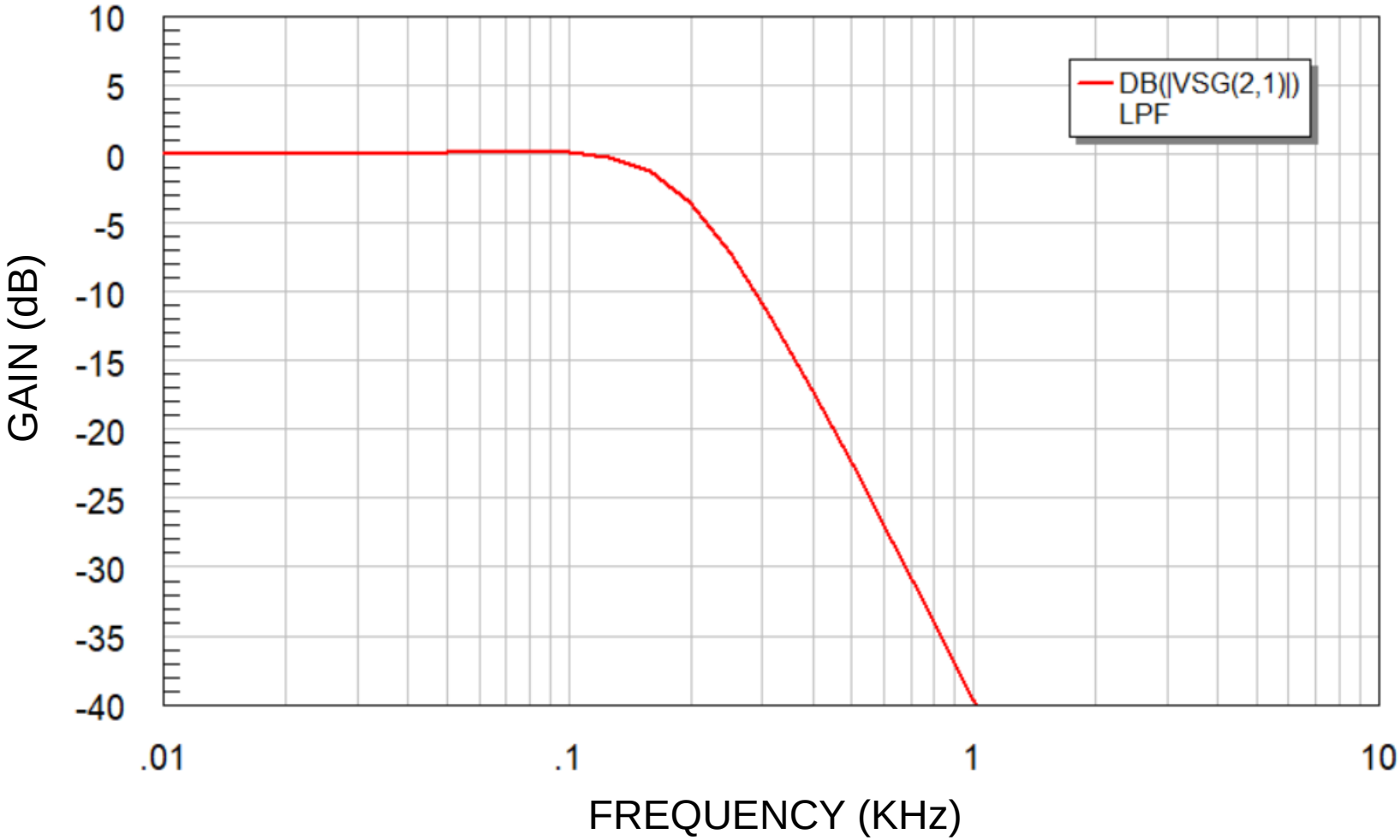
IN SHACK TELEMETRY RECEIVER

- Arduino provides DA conversion and TCP/IP LAN communication
- Arduino DA output is PWM waveform (976 Hz)
- Active 5th order LPF
- Bias circuit with negative voltage source
- Bias circuit provides +8v for Arduino
- Non-inverting buffer provides gain adjustment
- Max voltage clamp at +10v for LP-500

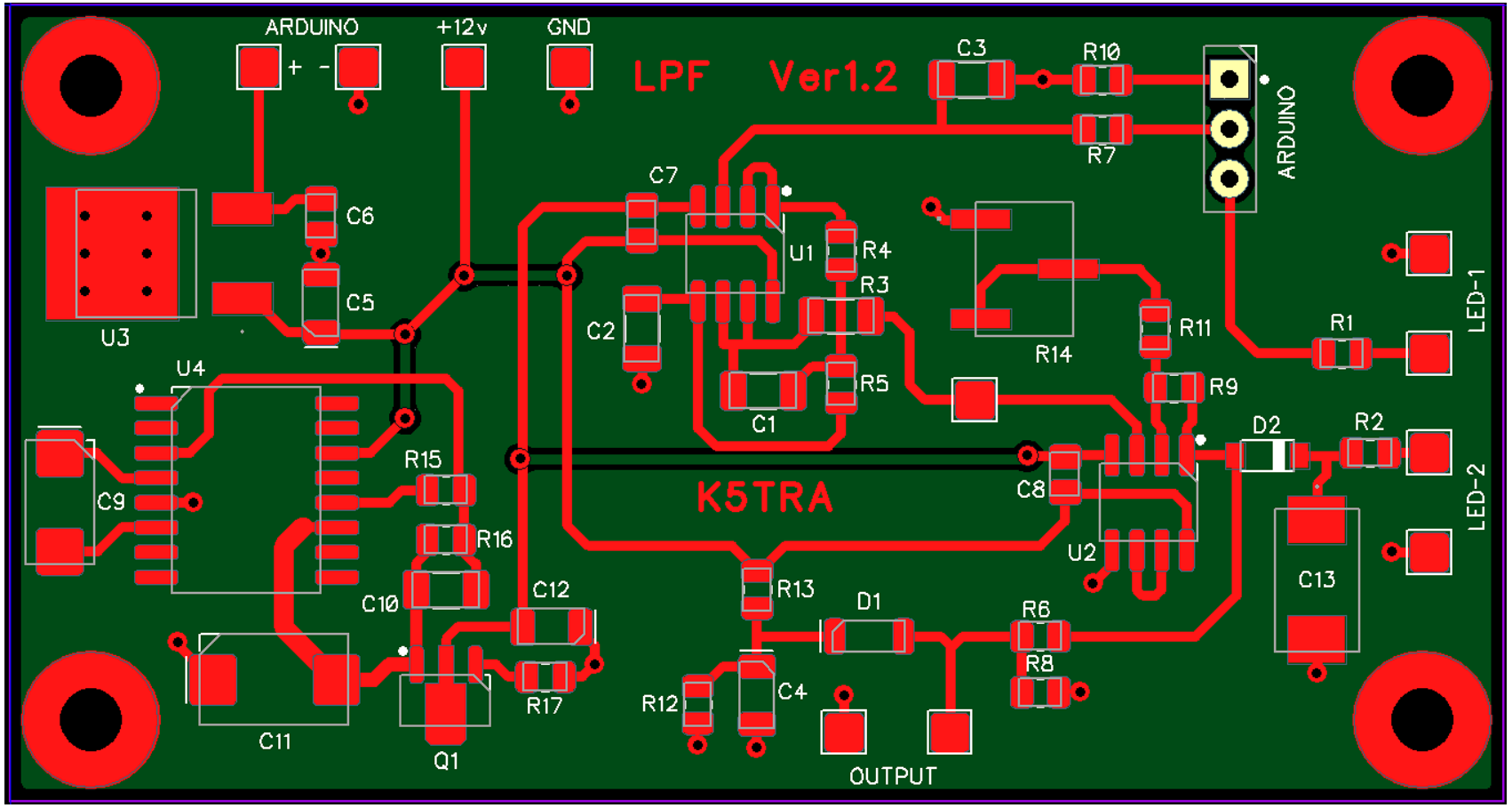
TELEMETRY RECEIVER BOARD SCHEMATIC



LOWPASS FILTER RESPONSE



TELEMETRY RECEIVER BOARD



ARDUINO CLIENT PROGRAMMING

```
/*
 * K5TRA  -  Remote Client
 * Power Sensor Analog Interface
 * 19 April  2022
 *
 */

// ARDUINO #1: TCP CLIENT + AnalogIO input
#include <AnalogIO.h>
#include <SPI.h>
#include <Ethernet.h>

AnalogIn potentiometer(A2, BIT8); // sensor is attached to pin A2
unsigned short valuePotentiometer = 0;
const int serverPort = 4080;

byte mac[] = { 0xA8, 0x61, 0x0A, 0xAE, 0x75, 0x0E };
IPAddress clientAddress(192, 168, 0, 56);
IPAddress serverAddress(192, 168, 0, 55);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress myDns(8, 8, 8, 8);

EthernetClient TCPclient;
```

```
void setup() {
  Serial.begin(9600);

  // initialize the Ethernet shield using the static IP address:
  Ethernet.begin(mac, clientAddress, myDns, gateway, subnet);

  Serial.println("ARDUINO #1: TCP CLIENT + AnalogIO input");
  // connect to TCP server (Arduino #2)
  if (TCPclient.connect(serverAddress, serverPort))
    Serial.println("Connected to TCP server");
  else
    Serial.println("Failed to connect to TCP server");
}


void loop() {
  /* Read the input values, write them on Serial Monitor. */
  valuePotentiometer = potentiometer.read();
  Serial.print("potentiometer: ");
  Serial.println(valuePotentiometer);
  TCPclient.write(valuePotentiometer); // Send input values to server
  TCPclient.flush();
  // connect to TCP server (Arduino #2)
  if (!TCPclient.connect(serverAddress, serverPort))
    Serial.println("Failed to connect to TCP server");
}
```

ARDUINO SERVER PROGRAMMING

```
/*
 * K5TRA  -  Server (with display)
 * Power Sensor Analog Interface
 * 19 April  2022
 *
 */

// ARDUINO #2: TCP SERVER + AnalogIO output
#include <AnalogIO.h>
#include <SPI.h>
#include <Ethernet.h>

AnalogOut led(6, BIT8);
unsigned short valueWrite = 10;
unsigned short valueWritten;
const int serverPort = 4080;

byte mac[] = { 0xA8, 0x61, 0x0A, 0xAE, 0x75, 0x0C };
IPAddress serverAddress(192, 168, 0, 55);
IPAddress gateway(192, 168, 0, 1);
IPAddress subnet(255, 255, 255, 0);
IPAddress myDns(8, 8, 8, 8);

EthernetServer TCPserver(serverPort);
```

```
void setup() {
   Serial.begin(9600);

   // initialize the Ethernet shield using the static IP address:
   Ethernet.begin(mac, serverAddress, myDns, gateway, subnet);
   Serial.println("ARDUINO #2: TCP SERVER + AnalogIO output");
   TCPserver.begin();   // Listening for a TCP client (from Arduino #1)
}

void loop() {
  // Wait for a TCP client from Arduino #1:
    EthernetClient client = TCPserver.available();

  if (client) {
   // Read the command from the TCP client:
   valueWrite = client.read();
   Serial.print("- Received command: ");
   Serial.println(valueWrite);
   valueWritten = led.write(valueWrite);
   Serial.print(" voltage = ");
   Serial.println(valueWritten*5.0/255);
  }
}
```

SUMMARY

- Five band PA output sensors
- Level summer and Arduino AD conversion
- Arduino ethernet shield board for TCP/IP
- In radio shack Arduino DA conversion
- LPF filtered PWM output from Arduino
- LP-500 display