GENERAL

arduino frequency display for kenwood ts 520s hf ham radio

Published: September 19, 2026 | Index ID: #-

Arduino Frequency Display for Kenwood TS 520S HF Ham Radio

Ham radio enthusiasts always look for ways to enhance their rigs with modern electronics. One of the most popular upgrades is adding a real time frequency display that works seamlessly with the Kenwood TS 520S HF transceiver. The TS 520S, a versatile 1 /kW HF transceiver, already offers a clear LCD screen, but many operators prefer a dedicated, larger display or an external meter that can be positioned for easier viewing. With an Arduino microcontroller, hobbyists can build a custom frequency counter that taps into the TS 520S's **frequency synthesizer output** and shows the current operating frequency on an external screen or even logs it to a computer.

This article walks you through the entire process: from understanding the TS 520S's frequency output, selecting the right Arduino board and display, wiring the circuit, writing the code, and finally testing and troubleshooting. Whether you're a beginner looking to dip your toes into DIY radio projects or an experienced ham who wants to fine tune a professional setup, the guide below will help you create a reliable, accurate, and eye catching frequency display.

Why Build an Arduino Frequency Display?

While the TS 520S's built in LCD is adequate for most purposes, there are several compelling reasons to add a separate frequency display:

- Visibility: A larger, high contrast screen can be positioned in a more ergonomic location.
- Customization: You can choose the display format—seven segment, OLED, or even a web interface.
- Data Logging: With Arduino's serial capabilities, you can log frequencies, timestamps, and other parameters to a PC or SD card.
- Integration: Combine the display with other DIY modules—e.g., a power meter, temperature sensor, or a remote control interface.
- Cost Effectiveness: Using readily available Arduino components keeps the project affordable compared to commercial frequency counters.

Understanding the Kenwood TS 520S Frequency Output

The TS 520S's internal synthesizer generates the operating frequency and outputs a 10 /MHz reference that can be tapped for external measurement. The key pin for this purpose is the **"FREQ OUT"** pin located on the front panel. It emits a square wave whose frequency is directly proportional to the transceiver's current operating frequency.

To convert the transceiver's frequency to a readable value, the Arduino must:

- Read the square wave from the FREQ OUT pin.
- Count the pulses over a defined time interval.
- Calculate the frequency using the known time base.
- Display the result on the chosen screen.

Because the TS 520S outputs a high frequency square wave (typically up to 30 MHz), it is essential to attenuate the signal to a level safe for the Arduino's input pins (maximum 5 /V).

Choosing the Right Arduino Board

Several Arduino boards can handle this project. The choice depends on display type, available I/O, and your familiarity with the platform.

Arduino Uno

The classic Uno is a solid choice for beginners. It offers 14 digital I/O pins, 6 analog inputs, and a 16 MHz clock. It is more than sufficient for a simple frequency counter and can drive a 16 segment LCD or a small OLED.

Arduino Nano

For a compact build, the Nano provides the same functionality in a smaller footprint. Its pinout is similar to the Uno, making the code largely interchangeable.

Arduino Mega

If you plan to expand the project (e.g., adding an SD card logger or multiple displays), the Mega's 54 digital I/O pins and 4 analog inputs give you plenty of headroom.

Selecting the Display Module

The display choice depends on visual preference and the amount of information you want to show. Below are three popular options:

Seven Segment Display

- Ideal for a minimalistic look.
- Can display up to 7 digits, sufficient for 10 /MHz frequencies.
- Requires a driver IC (e.g., MAX7219) to handle multiple segments.

OLED (0.96 3 128×64)

- High contrast, low power consumption.
- Can show text, numeric values, and simple graphics.
- Uses I²C communication, requiring only two pins.

LCD (16×2 Character)

- Commonly used in ham radio projects.
- Easy to drive with the LiquidCrystal library.
- Can show frequency as a string with optional status indicators.

For this guide, we'll focus on an OLED display because of its clarity and ease of use. The same principles apply to other display types with minor code adjustments.

Components List

- Kenwood TS 520S HF Transceiver
- Arduino Uno (or Nano/Mega)
- 0.96 3 128×64 I²C OLED display
- Resistors: 10 /k:'Â ö³© (for voltage divider)
- Capacitors: 0.1 /µF (filtering)
- Voltage regulator (if needed)
- Connecting wires
- Breadboard or perfboard for prototyping

- Optional: 10 /k:’ VÆÂ W &W6—7F÷" `or I²C lines

Circuit Design and Wiring

Attenuating the Frequency Output

The TS 520S’s FREQ OUT pin can produce a square wave up to 30 MHz. Arduino input pins cannot handle such high frequencies or voltages. We’ll use a simple resistor divider to bring the voltage down to 5 V and a capacitor to filter the signal.

Here’s a recommended attenuation network:

- Resistor 1: 10 /k:•
- Resistor 2: 1 /k:•
- Capacitor (C1): 0.1 /µF (parallel to Resistor 2)

Connect the FREQ OUT pin to the junction between Resistor 1 and Resistor 2. The output of Resistor 2 (after the capacitor) goes to the Arduino digital pin (e.g., D2). The other side of Resistor 2 connects to ground.

Connecting the OLED Display

The OLED uses I²C, so connect:

- OLED VCC !’ Arduino 5 /V
- OLED GND !’ Arduino GND
- OLED SCL !’ Arduino A5 (Uno) / D1 (Nano)
- OLED SDA !’ Arduino A4 (Uno) / D0 (Nano)

Optionally add 10 /k:’ VÆÂ W &W6—7F÷'2 &WGpeen VCC and SCL/SDA to improve signal integrity.

Software Setup

Installing the Required Libraries

Open the Arduino IDE and install the following libraries via the Library Manager:

1. Adafruit SSD1306 (for the OLED)
2. Adafruit GFX (graphics library)

These libraries provide high level functions for drawing text and graphics on the OLED.

Code Overview

The main tasks the sketch performs are:

- Initialize the OLED display.
- Configure the frequency input pin as an interrupt.
- Count pulses over a 100 ms window.
- Calculate the frequency in Hz.
- Display the result on the OLED.

Below is the complete sketch with detailed comments.

Arduino Sketch

\n#include <Wire.h>\n#include <Adafruit_GFX.h>\n#include <Adafruit_SSD1306.h>\n#define SCREEN_WIDTH 128 // OLED display width, in pixels\n#define SCREEN_HEIGHT 64 // OLED display height, in pixels\n#define OLED_RESET -1 // Reset pin # (or -1 if sharing Arduino reset)\nAdafruit_SSD1306

display(SCREEN_WIDTH, SCREEN_HEIGHT, &Wire, OLED_RESET);\nconst uint8_t freqPin = 2; // Digital pin connected to attenuated FREQ OUT\nvolatile uint32_t pulseCount = 0; // Pulse counter (volatile for ISR)\nunsigned long lastReadTime = 0; // Timestamp of last frequency calculation\nfloat frequency = 0.0; // Calculated frequency in Hz\nvoid IRAM_ATTR countPulse() { // ISR for counting pulses\n pulseCount++;\n}\nvoid setup() {\n Serial.begin(115200);\n pinMode(freqPin, INPUT_PULLUP); // Input with internal pull up (not strictly needed)\n attachInterrupt(digitalPinToInterrupt(freqPin), countPulse, RISING); // Count rising edges\n // Initialize OLED\n if(!display.begin(SSD1306_SWITCHCAPVCC, 0x3C)) { // I2C address 0x3C\n Serial.println(F("SSD1306 allocation failed"));\n for(;;); // Loop forever\n }\n display.clearDisplay();\n display.setTextSize(2);\n display.setTextColor(SSD1306_WHITE);\n display.setCursor(0,10);\n display.println(F("Freq:"));\n display.display();\n}\nvoid loop() {\n unsigned long currentTime = millis();\n if(currentTime - lastReadTime >= 100) { // 100 ms measurement window\n detachInterrupt(digitalPinToInterrupt(freqPin)); // Stop counting during calculation\n noInterrupts();\n uint32_t count = pulseCount; // Copy the count\n pulseCount = 0; // Reset counter for next window\n interrupts();\n attachInterrupt(digitalPinToInterrupt(freqPin), countPulse, RISING); // Resume counting\n // Calculate frequency: count per 100 ms => multiply by 10 for per second\n frequency = count * 10.0; // Hz\n lastReadTime = currentTime;\n // Display the frequency\n display.clearDisplay();\n display.setTextSize(2);\n display.setTextColor(SSD1306_WHITE);\n display.setCursor(0,10);\n display.print(F("Freq: "));\n display.println((long)frequency, 0);\n display.display();\n // Optional: print to Serial for debugging\n Serial.print(F("Frequency: "));\n Serial.print(frequency);\n Serial.println(F(" Hz"));\n }\n}

Explanation of Key Sections

Baca Juga (Artikel Terkait)

- **advanced engineering electromagnetics solution manual edition hardcover by balanis constantine a published by wiley**
  URL: http://djjsearch.com/advanced-engineering-electromagnetics-solution-manual-edition-hardcover-by-balanis-constantine-a-published-by-wiley.pdf
- **advanced engineering electromagnetics solutions manual**
  URL: http://djjsearch.com/advanced-engineering-electromagnetics-solutions-manual.pdf
- **advanced engineering electromagnetics wiley 1989 grading**
  URL: http://djjsearch.com/advanced-engineering-electromagnetics-wiley-1989-grading.pdf
- **advanced engineering fluid mechanics by biswas**
  URL: http://djjsearch.com/advanced-engineering-fluid-mechanics-by-biswas.pdf