```
#include <Wire.h>
#include <Adafruit_MCP4725.h>

// Initialize the MCP4725 DAC
Adafruit_MCP4725 dac;

// Define constants for I2C and CI-V settings
const int MCP4725_ADDR = 0x60;  // I2C address of MCP4725
const long CI_V_BAUD = 9600;    // CI-V baud rate for IC-705
const int CI_V_BUFFER_SIZE = 11;  // Size of CI-V message buffer

// Variables to store frequency and calculated analog voltage
unsigned long frequency = 0;
float voltage = 0.0;

// Band-to-voltage mapping (example, you can adjust based on the XPA125B)
struct BandMapping {
  unsigned long minFreq;
  unsigned long maxFreq;
  float voltage;
};

BandMapping bands[] = {
  { 800000, 2999999, 0.23 },     // 160m band, 1.8 MHz to 2 MHz
  { 3000000, 4499999, 0.46 },    // 80m band, 3.5 MHz to 4 MHz
  { 4500000, 5999999, 0.69 },    // 60m band, 14 MHz to 14.35 MHz
  { 6000000, 8999999, 0.92 },    // 40m band, 7 MHz to 7.3 MHz
  { 9000000, 12999999, 1.15 },   // 30m band, 10.1 to 10.15 Mhz
  { 13000000, 16999999, 1.38 },  // 20m band, 14 MHz to 14.35 MHz
  { 17000000, 19999999, 1.61 },  // 17m band, 18.068 MHz to 18.168 MHz
  { 20000000, 22999999, 1.84 },  // 15m band, 21 MHz to 21.45 MHz
  { 23000000, 26999999, 2.07 },  // 12m band, 24.89 to 24.99 MHz
  { 27000000, 39999999, 2.30 },  // 10m band, 28 MHz to 29.7 MHz
  { 40000000, 60000000, 2.53 }   // 6m band, 51 Mhz to 54.0 Mhz
};

// Function to convert BCD to decimal
unsigned long bcdToDecimal(byte *bcdBytes, int length) {
  unsigned long result = 0;
  for (int i = 0; i < length; i++) {
    result = result * 100 + ((bcdBytes[i] >> 4) * 10) + (bcdBytes[i] & 0x0F);
  }
  return result;
}
```

```
// Function to calculate the analog voltage for a given frequency
float calculateVoltage(unsigned long freq) {
  for (int i = 0; i < sizeof(bands) / sizeof(BandMapping); i++) {
    if (freq >= bands[i].minFreq && freq <= bands[i].maxFreq) {
      return bands[i].voltage;
    }
  }
  return 0.0;  // Default if no band is matched
}

void setup() {
  // Initialize the serial communication for CI-V and for debugging
  Serial.begin(CI_V_BAUD);  // Set baud rate to match IC-705 CI-V baud
  Serial.println("IC-705 CI-V Frequency Reader & MCP4725 Voltage Control");

  // Initialize the MCP4725 DAC
  dac.begin(MCP4725_ADDR);

  // Set the default voltage to 0
  dac.setVoltage(0, false);
}

void loop() {
  byte buffer[CI_V_BUFFER_SIZE];

  // Check if there is enough data available in the serial buffer
  if (Serial.available() >= CI_V_BUFFER_SIZE) {

    // Read the first byte to check if it's the expected header byte for a
frequency message
    if (Serial.peek() == 0xFE) {  // Typical start of CI-V message
      Serial.readBytes(buffer, CI_V_BUFFER_SIZE);

      // Check if it's a valid frequency message (based on the CI-V protocol)
      if (buffer[3] == 0xA4) {  // 0xA4 indicates a frequency message

        // Extract frequency bytes (assuming they are in BCD format from bytes 4-
8)
        byte freqBytes[] = { buffer[8], buffer[7], buffer[6], buffer[5] };

        // Convert BCD to decimal frequency
        frequency = bcdToDecimal(freqBytes, 4);

        // Valid frequency range check (e.g., 100 kHz to 60 MHz)
        if (frequency < 100000 || frequency > 60000000) {
```

```
          Serial.println("Invalid frequency detected! Skipping...");
          return;  // Skip processing if frequency is out of range
        }

        // Display the valid frequency on the serial monitor
        Serial.print("Freq: ");
        Serial.print(frequency);
        Serial.print(" Hz ");

        // Calculate the appropriate analog voltage based on the frequency
        voltage = calculateVoltage(frequency);
        int analogValue = (int)(voltage / 5.12 * 4095);  // MCP4725 uses 12-bit
DAC (0 to 4095)
        dac.setVoltage(analogValue, false);  // Set DAC output voltage

        // Display the calculated voltage
        Serial.print(" | Voltage: ");
        Serial.print(voltage);
        Serial.println(" V");
      } else {
        Serial.println("Invalid message type! Skipping...");
      }
    } else {
      // Skip and discard bytes that don't match the expected format
      Serial.read();  // Read and discard this invalid byte
    }
  }
}
```