

```

// August 23, 2025
#include <MCUFRIEND_kbv.h>
#include <Adafruit_GFX.h>
#include "arduinoFFT.h"

//***** TFT Color definitions etc*****
#define BLACK 0x0000
#define WHITE 0xFFFF
#define GREEN 0x07E0
#define RED 0xF800
#define GRAY 0x8410

const int graphTop = 5;
//const int graphHeight = 300;
const int graphHeight = 305;
//const int graphLeft = 5;
const int graphLeft = 35;
const int graphWidth = 384;
int xPos = graphLeft;
MCUFRIEND_kbv tft;

//*****important FFT definitions etc FFT is vers 1.5 not the latest *****
#define SAMPLES 256// Must be a power of 2
#define SAMPLING_FREQUENCY 250 // Hz
//#define SAMPLING_FREQUENCY 1000 // Hz
double vReal[SAMPLES];
double vImag[SAMPLES];
arduinoFFT FFT = arduinoFFT();

void drawAxes();
void clearGraphArea();

void setup()
{
  // Initialize analog input or generate a test signal
  // TFT
  uint16_t ID = tft.readID();
  tft.begin(ID);
  tft.setRotation(1);
  tft.fillScreen(BLACK);
}

```

```

void loop()
{
// Acquire SAMPLES data points into vReal
// For example, reading from an analog pin:
int y;// height of frequency component
for (int i = 0; i < SAMPLES; i++) {
vReal[i] = analogRead(A6);
vImag[i] = 0.0; // Initialize imaginary part to 0
delayMicroseconds(1000000 / SAMPLING_FREQUENCY); // Adjust for sampling rate
}
FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);
double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING_FREQUENCY);
tft.setTextSize(2);
tft.setCursor(80, 10);
tft.print("peak frequency = ");
tft.print(peak);
tft.println(" Hz");

for (int i = 0; i < (SAMPLES / 2); i++)
{
y = map(vReal[i],40000,0, graphTop + graphHeight, graphTop); // changes the analog input range from 0
to 40000, to 310, 5
// Draw line from previous point
if (xPos > graphLeft)
{
//tft.drawLine(xPos, 0, xPos, y, GREEN);
tft.drawLine(xPos, graphTop + graphHeight, xPos,graphTop + graphHeight-y, GREEN);
}
xPos=xPos+3;//3*128 = 384 graphWidth
// Scroll graph when end is reached
if ((xPos >= graphLeft + graphWidth))
{
delay(2500);//allow time before update
clearGraphArea();
drawAxes();
xPos=graphLeft;
}
}

delay(1000); // Wait before next FFT
}

```

```

// Draw X and Y axes with labels and grid lines
void drawAxes()
{
  tft.setTextSize(1);
  tft.setTextColor(WHITE);

  // Y axis
  tft.drawLine(graphLeft, graphTop, graphLeft, graphTop + graphHeight, WHITE);//(20,40,20,40+160)

  for (int i = 0; i <= 4; i++)
  {
    int y = graphTop + i * (graphHeight / 4);
    tft.drawLine(graphLeft, y, graphLeft + graphWidth, y, GRAY); // horizontal grid
    tft.setCursor(0, y - 4);
    tft.print(100 - i * 25);
  }

  // X axis line
  tft.drawLine(graphLeft, graphTop + graphHeight, graphLeft + graphWidth, graphTop + graphHeight,
  WHITE);
  tft.setCursor(graphLeft + graphWidth + 5, graphTop + graphHeight - 8);
  tft.print("frequency");
}

// Clears the graph area only (not the whole screen)
void clearGraphArea()
{
  tft.fillRect(graphLeft + 1, graphTop + 1, graphWidth - 1, graphHeight - 1, BLACK);
}

```