```
// FIR Filter Implementation on Raspberry Pi Pico W
#define SAMPLE_RATE 20000  // Sampling frequency (Hz)
#define BUFFER_SIZE 151      // FIR filter order
#define ANOTHER_BUFFER 100 // Buffer for batch processing

// FIR filter coefficients generated from MATLAB
float h[BUFFER_SIZE] = {-0.0001969451123765 58,-0.
00068248608923588 1,-0.00020634690079446 2,0.00047733172753077 3,
0.00036345267821745 7,-7.7104613589309 3e-05,3.
78815165511894e-18,8.7780712874034 8e-05,-0.00047006450103214,
-0.00069841130039301 7,0.00033952333255085 8,0.00125338278922503,
0.00040021087293401 3,-0.00096914474455176 3,-0.
00076589023785716 5,0.00016727922544584 8,-8.1658252067820 7e-18,
-0.00019760653559224 2,0.00106829710580943,0.00159472372627213,
-0.00077578763522200 8,-0.00285651251019979,-0.
00090737057960226 9,0.00218141517342954,0.00170879642230349,-0.
00036951865535098 3,-1.2237945832622 7e-17,0.00042695868390885 1,
-0.00228147202344769,-0.00336580385659093,0.00161820197105458,
0.00588939614633983,0.00184954140434993,-0.00439739872202489,
-0.00340789566459 26,0.00072937814107226 7,-3.9848388033087 7e-18,
-0.00082671595238501,0.00437870877585076,0.00640644543865718,
-0.00305636350875138,-0.01104441072851 35,-0.00344587700390213,
0.00814458690715 95,0.00627885323991601,-0.00133770900689741,3.
18598747284536e-17,0.00150565968876661,-0.00795596231015279,-0.
01162234127753 83,0.00554097498859423,0.02002755703650 26,0.
00625630474599486,-0.01482116172706 29,-0.01146544380742 72,0.
00245425118336613,-6.7192408712576 9e-18,-0.00280066307003986,0.
01493837590191 88,0.02207139184014 76,-0.01066630635525 71,-0.
03917943847883 65,-0.01247498426913 14,0.03022806161033 77,0.
02401732633557 19,-0.00530674046301725,7.5444998063230 8e-18,0.
00658181194017162,-0.03709839629566 33,-0.05866448535014 78,0.
03086952161776 29,0.12650375610181 4,0.04660480210377 7,-0.
13866694607201,-0.15130427009047 1,0.06079177878354 87,0.
19991930036909 5,0.06069370433420 2,-0.15081630768574 8,-0.
13799630549128 1,0.04630428242664 42,0.12548390103892 3,0.
03057075978511 09,-0.05800167020650 63,-0.03661895468547 94,0.
```

```
0064860161214652,7.4223328646368e-18,-0.0052120719808562,0.
023549106649375,0.029588394691652,-0.012190064406081,-0.
038218364151109,-0.010386477049621,0.021454386755704,0.
014494831896782,-0.002712593413194,-6.4960307619545e-18,0.
002368319149481,-0.011043178606905,-0.014248050280758,0.
006002722904310,0.019177945945209,0.005295290130186,-0.
011084385406136,-0.007571977270146,0.001429964800476,3.
019305383145590e-17,-0.001264942390654,0.005924017745450,0.
007666761314800,-0.003236149597501,-0.010347502078659,-0.
002856534786441,0.005972687079894,0.004071858141714,-0.
000766782148306,-3.686125371741e-18,0.000672868013850299,
-0.003135126610878,-0.004033946324388,0.001691756606102,0.
005371045837009,0.001471332787693,-0.003050946795520,-0.
002061620125383,0.000384601049469,-1.098851735641e-17,
-0.000330745133012,0.001524623993503,0.001940163095540,
-0.000804520510600,-0.002525121808283,-0.
000683824061525,0.001401938689555,0.000936904564869,-0.
000172949707664,-7.135673239278e-18,0.000146032257326,
-0.000668443659431,-0.000846392532943,0.
000350133436693,0.001099910668784,0.000299319348801,-0.
000619617895714,-0.000420491799140,7.933952992522e-05,3.
466899755177e-18,-7.160350022559e-05,0.000343150640596,0.
000458931074833,-0.000202267953251};


// Circular buffer for past samples
float buffer[BUFFER_SIZE] = {0};

// Function to implement FIR filtering
float fir_filter(float new_sample) {
    // Shift buffer left
    for (int i = 0; i < BUFFER_SIZE - 1; i++) {
        buffer[i] = buffer[i + 1];
    }
    buffer[BUFFER_SIZE - 1] = new_sample;
```

```
    // Compute FIR convolution
    float output = 0.0;
    for (int i = 0; i < BUFFER_SIZE; i++) {
        output += buffer[i] * h[i];
    }
    return output;
}

void setup() {
    Serial.begin(115200);

    // Initialize ADC (Microphone Input)
    analogReadResolution(12);  // 12-bit ADC (0-4095)
    pinMode(A0, INPUT);  // Use A0 for microphone input

    // Initialize PWM for Output (D9)
    pinMode(9, OUTPUT);
    analogWriteResolution(12); // 12-bit PWM output (0-4095)
}

void loop() {
    float processing_buffer[ANOTHER_BUFFER];

    // Read 1000 samples and store them in processing_buffer
    for (int i = 0; i < ANOTHER_BUFFER; i++) {
        int raw_adc = analogRead(A0);
        processing_buffer[i] = ((float)raw_adc / 4095.0) * 2.0
- 1.0; // Normalize to [-1,1]
        delayMicroseconds(1000000 / SAMPLE_RATE); // Maintain
sampling rate
    }

    // Process 1000 stored samples one by one
    for (int i = 0; i < ANOTHER_BUFFER; i++) {
        float output_signal = fir_filter(processing_buffer[i]
```

```
        // Convert back to PWM (0 - 4095)
        int pwm_output = (int)((output_signal + 1.0) * 2047.5
        analogWrite(9, pwm_output);

        // Display the filtered signal in the Serial Plotter
        Serial.println(output_signal);
        delayMicroseconds(1000000 / SAMPLE_RATE);
    }
}
```