

TECHNOLOGICAL INSTITUTE OF THE PHILIPPINES – MANILA

**ANITU: A MACHINE LEARNING BASED DEVICE WITH SOUND RECOGNITION AND
NOTIFICATION SYSTEM FOR FOREST CONSERVATION**



**ADVANCED EMBEDDED SYSTEMS
MEMC 122-MCPE11S1**

SUBMITTED BY:

**CEZAR, ALQUIN
GALVEZ, JOHN ROY**

SUBMITTED TO:

ENGR. RUFO I. MARASIGAN JR.

May 5, 2022

Arduino Code

```
/*
  File/Sketch Name: AudioFrequencyDetector
*/

#include "arduinoFFT.h"

#define SAMPLES 128          //SAMPLES-pt FFT. Must be a base 2 number. Max 128 for Arduino Uno.
#define SAMPLING_FREQUENCY 2048 //Ts = Based on Nyquist, must be 2 times the highest expected frequency.

arduinoFFT FFT = arduinoFFT();
unsigned int samplingPeriod;
unsigned long microSeconds;
double vReal[SAMPLES]; //create vector of size SAMPLES to hold real values
double vImag[SAMPLES]; //create vector of size SAMPLES to hold imaginary values
void setup()
{
  Serial.begin(9600); //Baud rate for the Serial Monitor
  samplingPeriod = round(1000000*(1.0/SAMPLING_FREQUENCY)); //Period in microseconds
}

void loop()
{
  /*Sample SAMPLES times*/
  for(int i=0; i<SAMPLES; i++)
  {
    microSeconds = micros(); //Returns the number of microseconds since the Arduino board began running the
    current script.

    vReal[i] = analogRead(0); //Reads the value from analog pin 0 (A0), quantize it and save it as a real term.
    vImag[i] = 0; //Makes imaginary term 0 always

    /*remaining wait time between samples if necessary*/
    while(micros() < (microSeconds + samplingPeriod))
    {
      //do nothing
    }
  }
  for(int i = 0; i < sizeof(vReal); i++)
  {
    Serial.print(vReal[i]+vImag[i]);
    Serial.print(" ");
  }
  /*Perform FFT on samples*/
  FFT.Windowing(vReal, SAMPLES, FFT_WIN_TYP_HAMMING, FFT_FORWARD);
  FFT.Compute(vReal, vImag, SAMPLES, FFT_FORWARD);
  FFT.ComplexToMagnitude(vReal, vImag, SAMPLES);

  /*Find peak frequency and print peak*/
  double peak = FFT.MajorPeak(vReal, SAMPLES, SAMPLING_FREQUENCY);
  //Serial.print("Hz: ");

  Serial.print(";");
  Serial.print(peak); //Print out the most dominant frequency.
  delay(3000);
}
```

Python Code

```
#!/usr/bin/env python3
import serial
import numpy as np
import time
import board
import random
import requests
import json
from datetime import date

ser = serial.Serial('/dev/ttyACM0', 9600, timeout=1)
ser.reset_input_buffer()
samples = [] #Samples in Number Format

def convert(someList):
    for sample in someList:
        try:
            samples.append(float(sample))
        except:
            pass
    return samples

def postData(sample,average, hz):
    # post data using request. Other option is using http client
    # Change the IP address, use your own computer local IP address
    url = "http://192.168.100.43:5000/api/sound/add"
    payload = json.dumps({
        "Samples": (sample),
        "Average": (average),
        "Hertz": (hz),
        "date": time.strftime("%Y-%m-%d %H:%M:%S")
    })
    headers = {
        'Content-Type': 'application/json'
    }
    response = requests.request("POST", url, headers=headers, data=payload)
    print(response.text)

while True:
    try:
        if ser.in_waiting > 0:
            line = ser.readline().decode('utf-8').rstrip()
            sample, hz = line.split(';')
            sample = sample.split(' ')
            hz = float(hz)
            #print(sample)
            #print("\n")
            #average = np.maximum(convert(sample))
            #average = float(average)
            average = sample[0]
            print("\nAverage: ",average)
            print("\nHertz: ', hz)
            postData(sample,average, hz)

            print("\nDatabase Updated")
            time.sleep(5.0)
        except RuntimeError as error:
            continue
        except Exception as error:
            continue
```

server.js

```
const express = require('express');
const app = express();
const sound = require('./routes/api/sound')

const cors = require('cors')

app.use(cors({
  origin: 'http://192.168.100.43:5000',
  credentials: true
}));

app.use(express.json({ extended: false }));
// app.get('/', (req, res) => res.send('API Running'));
app.get('/', function (req, res) {
  res.header("Access-Control-Allow-Origin", "*");
  res.send('API Running: Final Exam');
})

const PORT = process.env.PORT || 5000;

app.use('/api/sound',sound)
app.listen(PORT, () => console.log(`Server started on port ${PORT}`));
```

db.js

```
var mysql = require('mysql');

var connection = mysql.createConnection({
  host: 'localhost',
  user: 'pi',
  password: 'raspberry',
  database: 'FinalExam',
});

connection.connect(function (error) {
  if (!!error) {
    console.log(error);
  } else {
    console.log('MySQL Database Connected...!');
  }
});

module.exports = connection;
```

sounds.js

```
var express = require('express');
var router = express.Router();
var dbConn = require('../config/db');

//ADD
router.post('/add',(req, res) =>{
  // get the input from the user trough request (req)
  console.log(req.body);
  var Samples = req.body.Samples;
  var Average = req.body.Average;
  var Hertz = req.body.Hertz;
  var date = req.body.date;
  // connect to mysql database and perform INSERT Query
  sqlQuery = `INSERT INTO sound_tb (Samples, Average, Hertz, date) VALUES`
  ('${Samples}','${Average}','${Hertz}','${date}')`
  dbConn.query(sqlQuery, function( error, results, fields ){
    if (error) throw error;
    res.status(200).json(results);
  });
});

// INSERT
// Inserting entry between rows given an ID
router.post('/insert/:id',(req, res) =>{
  // get the input from the user trough request (req)
  console.log(req.body);
  var SignalID = req.body.SignalID;
  var Samples = req.body.Samples;
  var Average = req.body.Average;
  var Hertz = req.body.Hertz;
  var date = req.body.date;
  // connect to mysql database and perform INSERT Query
  sqlQuery = `INSERT INTO sound_tb (SignalID, Samples, Average, Hertz,date) VALUES`
  ('${SignalID}','${Samples}','${Average}','${Hertz}','${date}')`
  dbConn.query(sqlQuery, function( error, results, fields ){
    if (error) throw error;
    res.status(200).json(results);
  });
});

// VIEW
// VIEWING OF ALL RECORDS
// GET /api/sound/view
router.get('/view', (req, res) => {
  dbConn.query('SELECT * FROM sound_tb', function (error, results,fields ) {
    if (error) throw error;
    res.status(200).json(results);
  });
});

//VIEWING DEPENDING ON ID
// GET /api/sound/view/id
router.get('/view/:id',(req,res)=> {
  dbConn.query('SELECT * FROM sound_tb WHERE SignalID = ?',[req.params.id] ,(error, results,fields)=>{
    if (error) throw error;
    res.status(200).json(results);

  })
});
```

```

// UPDATE
// PUT /api/sound/update/id
router.put('/update/:id',(req, res) =>{
  // get the input from the user trough request (req)
  console.log(req.body);
  var Samples = req.body.Samples;
  var Average = req.body.Average;
  var Hertz = req.body. Hertz;
  var date = req.body. date;
  // connect to mysql database and perform update Query
  sqlQuery = `UPDATE sound_tb SET Samples = '${Samples}', Average = '${Average}', Hertz = '${Hertz}', date =
'${date}' WHERE SignalID = ?`
  dbConn.query(sqlQuery, [req.params.id],( error, results, fields )=>{
    if (error) throw error;
    res.status(200).json(results);
  });
});

// DELETE
// DELETE /api/sound/delete/id
router.delete('/delete/:id',(req,res)=> {
  dbConn.query('DELETE FROM sound_tb WHERE SignalID = ?', [req.params.id],(error, results,fields)=>{
    if (error) throw error;
    res.status(200).json(results);

  })
});

module.exports = router;

```

Frontend Files

main.js

```

import { createApp } from 'vue'
import App from './App.vue'
import router from './router'

import LineChart from './components/LineChart.vue';

createApp(App).component('LineChart', LineChart).use(router).mount('#app')

```

App.vue

```

<template>
<div>
  <nav>
    <router-link to="/">Home</router-link> |
    <router-link to="/application">Application</router-link> |
    <router-link to="/about">About</router-link>
  </nav>
  <router-view/>
</div>
</template>

```

```
<style>
#app {
  font-family: Avenir, Helvetica, Arial, sans-serif;
  -webkit-font-smoothing: antialiased;
  -moz-osx-font-smoothing: grayscale;
  text-align: center;
  color: #2c3e50;
}

nav {
  padding: 50px;
  list-style-type: none;
  margin: 0;
  padding: 10px;
  overflow: hidden;
  background-color: rgb(0, 255, 128);
}

nav a {
  font-weight: bold;
  color: #000000;
}

nav router-link:hover:not(.active){
  background-color:#111 ;
}
nav a.router-link-exact-active {
  color: #42b983;
  font-size: 20px
}
</style>
```

COMPONENTS

HelloWorld.vue

```
<template>
  <div class="hello">
    <LineChart/>
  </div>
</template>

<script>

export default {
  name: 'HelloWorld',
  props: {
    msg: String
  }
}
</script>

<!-- Add "scoped" attribute to limit CSS to this component only -->
<style scoped>
h3 {
  margin: 40px 0 0;
}
ul {
```

```
list-style-type: none;
padding: 0;
}
li {
display: inline-block;
margin: 0 10px;
}
a {
color: #42b983;
}
</style>
```

LineChart.vue

```
<template>
<div class="lineChart">
  <div id="chartContainer" style="margin: 30px">
    <canvas id="myChart"></canvas>
  </div>
</div>
</template>

<script>
import Chart from "chart.js/auto";
import { ref, computed } from "vue";
import axios from "axios";

let Average = [];
let Hertz = [];
let historyDates = [];
let date = [];
let myChart;
myChart;

export default {
  name: "LineChart",
  mounted() {
    this.createDuration();
    this.updateSensorChart();
  },
  setup() {
    let updateSensorChart = () => {
      const ctx = document.getElementById("myChart");

      const labels = date;
      const data = {
        labels: labels,
        datasets: [
          {
            label: "First Value of 128 SAMPLES",
            data: Average,
            fill: false,
            borderColor: "rgb(0,0,255)",
            tension: 0,
            options: {
              responsive: true,
              maintainAspectRatio: false,
            },
          },
          {
            label: "Converted to Hertz",
```



```

    data: Hertz,
    fill: false,
    borderColor: "rgb(255,0,0)",
    tension: 0,
    options: {
      responsive: true,
      maintainAspectRatio: false,
    },
  },
],
};

//let myChart = null;

const chartWithKey = Chart.getChart("myChart");
if (chartWithKey !== undefined) {
  chartWithKey.destroy();
}

myChart = new Chart(ctx, {
  type: "line",
  data: data,
});
};

let readingsAPI = computed(() => {
  return "http://192.168.100.43:5000/api/sound/view";
});

console.log(readingsAPI.value);

let createDuration = (dateRange) => {
  axios.get(readingsAPI.value).then((response) => {
    historyDates = response;

    for (const property in historyDates.data) {
      let arrAverage = historyDates.data[property]["Average"];
      let arrHertz = historyDates.data[property]["Hertz"];
      let arrDate = historyDates.data[property]["date"];

      let d = new Date(arrDate);
      const month = d.getMonth();
      const day = d.getDate();
      const hours = d.getHours();
      const minute = d.getMinutes();
      const mSeconds = d.getSeconds();

      arrDate = month + "/" + day + " " + hours + ":" + minute + ":" + mSeconds;
      //console.log(arrDate)
      Average.unshift(arrAverage);
      Hertz.unshift(arrHertz);
      date.unshift(arrDate);
    }
    Average.reverse();
    Hertz.reverse();
    date.reverse();

    console.log(Average);
    console.log(Hertz);
    console.log(date);

    dateRange = Date;

```

```
});

    updateSensorChart();
};

return {
    updateSensorChart,
    createDuration,
};
},
};
</script>
```

index.js

```
import { createRouter, createWebHistory } from 'vue-router'
import Application from '../views/Application.vue'
import Home from '../views/Home.vue'

const routes = [
  {
    path: '/',
    name: 'home',
    component: Home
  },
  {
    path: '/application',
    name: 'application',
    component: Application
  },
  {
    path: '/about',
    name: 'about',
    // route level code-splitting
    // this generates a separate chunk (about.[hash].js) for this route
    // which is lazy-loaded when the route is visited.
    component: () => import(/* webpackChunkName: "about" */ '../views/AboutView.vue')
  }
]

const router = createRouter({
  history: createWebHistory(process.env.BASE_URL),
  routes
})

export default router
```

PAGES

Home.js

```
<template>
<div>
  <div class="header">
    <h4>
      ANITU: A MACHINE LEARNING BASED DEVICE WITH SOUND RECOGNITION AND
      NOTIFICATION SYSTEM FOR FOREST CONSERVATION
    </h4>
  </div>

  <div class="row">
    <div class="leftcolumn">
      <div class="card">
        <h2>Introduction</h2>
        <p style="text-align: justify">
          During the pre – colonial period in the Philippines, the Filipino
          ancestors, or the Ninuno, worships gods and goddesses of nature. The
          ninuno perform rituals for different things such as winning wars,
          having a bountiful harvest, even protecting their homes. The
          ancestors create human-like figurines that resembles a little human
          being, or what they called taotao. These little human like figures
          are also called Anito. Anitu or Anito is part of the Filipino
          Folklore which refers to the ancient or nature spirits. It is said
          that these spirits are inhabit the figurines, and serves as
          protector of shrines, forests, or sacred places.
        </p>
        <p style="text-align: justify">
          The Philippine land is covered by dense forest. It is one of the
          main sources of living and protecting the homes of the Pilipino
          people for many centuries from natural calamities such as
          landslides, flood, and devastating typhoons. The Philippines has
          abundant forestry, but also has a rapid population growth. Thus, the
          increase in demand of material for building houses.
        </p>
      </div>
      <div class="card">
        <h2>How to User the WebApp</h2>
        <p style="text-align: justify">
          This webapp aims to identify the hertz detected by the Sound Sensor
          integrated in the Arduino Uno R3 connected in Raspberry Pi 4 Model B
          serially. We perform Fast Fourier Transform (FFT) in 128 samples per
          upload where we use SAMPLING_FREQUENCY of 2048 which is based on
          Nyquist, must be 2 times the highest expected frequency. The
          arduinoFFT library in Arduino was able to identify the MarjoPeak in
          terms of Hertz which was verified using an app called Hertz
          Generator. The first sample on each run together with the equivalent
          hertz was then plotted in the graph in Application Page.
        </p>
      </div>
    </div>
    <div class="rightcolumn">
      <div class="card">
        <h2>Official Logo</h2>
        
        <p>As a Final Requirement for MEMC 122-MCPE11S1 - Advanced Embedded Systems</p>
      </div>
      <div class="card">
        <h3>Popular Post</h3>
        <div class="fakeimg">ANITU</div>
      </div>
    </div>
  </div>
</div>
```

```
        </div>
    </div>
</div>

<div class="footer">
    <h3>Footer</h3>
</div>
</div>
</template>
<style scoped>
* {
    box-sizing: border-box;
}

/* Add a gray background color with some padding */
body {
    font-family: Arial;
    padding: 10px;
    background: #f1f1f1;
}

/* Header/Blog Title */
.header {
    font-size: 30px;
    text-align: center;
    background: white;
}

/* Create two unequal columns that floats next to each other */
/* Left column */
.leftcolumn {
    float: left;
    width: 75%;
}

/* Right column */
.rightcolumn {
    float: left;
    width: 25%;
    padding-left: 20px;
}

/* Fake image */
.fakeimg {
    background-color: #aaa;
    width: 100%;
    padding: 20px;
}

/* Add a card effect for articles */
.card {
    background-color: white;
    padding: 20px;
}

/* Clear floats after the columns */
.row:after {
    content: "";
    display: table;
    clear: both;
}
```

```

/* Footer */
.footer {
  padding: 5px;
  text-align: center;
  background: #ddd;
  margin-top: 20px;
}

/* Responsive layout - when the screen is less than 800px wide, make the two columns stack on top of each other
instead of next to each other */
@media screen and (max-width: 800px) {
  .leftcolumn,
  .rightcolumn {
    width: 100%;
    padding: 0;
  }
}
</style>

```

Application.js

```

<template>
  <div class="row">
    <div class="column" style="width: 65%; height: 5vw">
      <div class="card">
        <div class="container" >
          <h2 style="text-align: left;">
            Using FFT to identify Max Peak of Sound Sensor
          </h2>
          <HelloWorld />
        </div>
      </div>
    </div>

    <div class="column" style="width: 35%; height: 5vw">
      <div class="card">
        <div class="container">
          <h3>General Problem</h3>
          <p>
            There has been an increasing report on illegal activities in the forest of the Philippines and authorities are
            having a hard time monitoring, recording, and tracking down different group. Furthermore, the illegal activities cannot
            be easily identified. In addition, forest rangers and authorities cannot secure the forest 24/7 which give the illegal
            loggers time to execute their activities.
          </p>
          <h3>General Objective</h3>
          <p>
            The goal of this project is to aid the authorities to cracked down illegal activities in the forest using unmanned
            device through sound recognition, also to reduce the risk of the Forest Rangers in patrolling forest.
          </p>
          <h3>Target Beneficiaries</h3>
          <p>
            Authorities,Forest Ranger, Community, Environment, Future Researchers
          </p>
        </div>
      </div>
    </div>
  </div>
</template>

<script>
// @ is an alias to /src
import HelloWorld from "@components/HelloWorld.vue";

```

```
export default {
  name: "HomeView",
  components: {
    HelloWorld,
  },
};
</script>

<style scoped>
p {
  text-align: justify;
}
img {
  border-radius: 50%;
}
* {
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}

/* Float columns side by side */
.column {
  float: left;
  width: 50%;
  padding: 20px 10px;
}

/* Remove extra left and right margins, due to padding in columns */
.row {
  margin: 0 -5px;
}

/* Clear floats after the columns */
.row:after {
  content: "";
  display: table;
  clear: both;
}

/* Style the counter cards */
.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2); /* this adds the "card" effect */
  padding: 16px;
  text-align: center;
  background-color: #f1f1f1;
}

/* Responsive columns - one column layout (vertical) on small screens */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
    display: block;
    margin-bottom: 50px;
  }
}
</style>
```

AboutView.js

<template>

```

<div class="row">
  <div class="column">
    <div class="card">
      
      <div class="container">
        <h2><b>Alquin Cezar</b></h2>
        <p><b>Position: </b>Instructor 1</p>
        <p><b>Degree: </b>Master of Engineering in Computer Engineering</p>
        <p><b>Organization: </b>Rizal Technological University - Boni</p>
        <p><b>Description: </b>An Instructor assigned at the College of Engineering, Architecture and Technology -
Computer Engineering Department. His especializations are Computer Hardware and Network subjects, such as
Printed Circuit Board Development, Logic Circuits, and Network 1 & 2. </p>
      </div>
    </div>
  </div>
  <div class="column">
    <div class="card">
      
      <div class="container">
        <h2><b>John Roy Galvez</b></h2>
        <p><b>Position: </b>Special Science Teacher 1</p>
        <p><b>Degree: </b>Master of Engineering in Computer Engineering</p>
        <p><b>Organization: </b>Camarines Sur National High School</p>
        <p><b>Description: </b>A secondary high school teacher teaching ICT related subjects in Grades 8 -11 as a
Return of
        Service in DOST-JLSS. His specialization focuses in Programming and Machine Learning. Curretly working
        with game developers in creating VR and 2D/3D application for the promotion of Bicol.</p>
      </div>
    </div>
  </div>
</div>
</template>
<style scoped>
p{
  text-align: justify;
}
img {
  border-radius: 50%;
}
* {
  box-sizing: border-box;
}

body {
  font-family: Arial, Helvetica, sans-serif;
}

/* Float four columns side by side */
.column {
  float: left;
  width: 50%;
  padding: 20px 10px;
}

/* Remove extra left and right margins, due to padding in columns */
.row {
  margin: 0 -5px;
}

/* Clear floats after the columns */
.row:after {
  content: "";

```

```
display: table;
clear: both;
}

/* Style the counter cards */
.card {
  box-shadow: 0 4px 8px 0 rgba(0, 0, 0, 0.2); /* this adds the "card" effect */
  padding: 16px;
  text-align: center;
  background-color: #f1f1f1;
}

/* Responsive columns - one column layout (vertical) on small screens */
@media screen and (max-width: 600px) {
  .column {
    width: 100%;
    display: block;
    margin-top: 20px;
    margin-bottom: 20px;
  }
}
</style>
```