**Test Plan - ECE 458, Spring 2020**

**Group 9 - Acoustic Awareness Enabler**

University of Massachusetts Dartmouth

|  |  |
| --- | --- |
| *Professor:* | Dr. Paul Fortier |
| *Customer:* | Dr. Karen Payton |
| *Advisor:* | Dr. Vance McCollough |

***­­­****Group Members:*

|  |  |
| --- | --- |
| Drew Martins: | Team Lead and Programmer |
| Stephen Felix: | Test Engineer |
| Ryan Dumont: | Electrical Design Engineer |
| Michael Benker: | | Systems Engineer |
| Jared Alves: | Signal Processing Engineer |

Due date: 2/27/2020

Table of Contents

[List of Figures 2](#_Toc33641572)

[List of Tables 2](#_Toc33641573)

[System Overview 3](#_Toc33641574)

[Background 3](#_Toc33641575)

[Project Objectives 3](#_Toc33641576)

[Concept of Operations 4](#_Toc33641577)

[Engineering Requirements 7](#_Toc33641578)

[Verification Cross Reference Matrix (VCRM) 7](#_Toc33641579)

[Test Plans Overview and Flowchart 8](#_Toc33641580)

[Test Plan Flow Chart 9](#_Toc33641581)

[Test Plan Verification Matrix 10](#_Toc33641582)

[Test Plans: Electronics 12](#_Toc33641583)

[Test Plans: Processing 28](#_Toc33641584)

[Test Plans: Coding 35](#_Toc33641585)

[Test Plans: Printed Circuit Board 40](#_Toc33641586)

[Test Plans: Enclosure 45](#_Toc33641587)

[Test Summary and Comments 48](#_Toc33641588)

[Updated Plan and Schedule 50](#_Toc33641589)

[ADC Test Code (C) 55](#_Toc33641590)

# List of Figures

[Figure 1 Design Concept 4](#_Toc33642622)

[Figure 2 Software Diagram 5](#_Toc33642623)

[Figure 3 Concept of Design 6](https://umassd-my.sharepoint.com/personal/mbenker_umassd_edu/Documents/ECE458%20Design%20Project/Test%20Plans/ECE-9%20Test%20Plans.docx#_Toc33642624)

[Figure 4 Device Example 6](#_Toc33642625)

[Figure 5 Test Plan Flow Chart 9](#_Toc33642626)

[Figure 6 Microphone Circuit 52](#_Toc33642627)

[Figure 7 Amplifier Test Plan Circuit 52](#_Toc33642628)

[Figure 8 Microphone and Amplifier Integration Circuit 53](#_Toc33642629)

[Figure 9 Digital Switch Testing Circuit 53](#_Toc33642630)

[Figure 10 Tentative Enclosure 54](#_Toc33642631)

# List of Tables

[Table 1 Customer Requirements 3](#_Toc33642678)

[Table 2 Verification Cross Reference Matrix 7](#_Toc33642679)

[Table 3 Test Plan Verification Matrix 10](#_Toc33642680)

[Table 4 Test Summary 48](#_Toc33642681)

# System Overview

## Background

The idea of the Acoustic Awareness Enabler is reported to have resulted from an incident in which the customer was watching a movie on her laptop with headphones. Her husband had called for her to respond, but she was unaware of this, her attention drawn to her computer. She thought then, what if there was a way for her headphones to let her know if someone was trying to get her attention? This was the apparent beginning of the Acoustic Awareness Enabler. Senior Design Project Group 9 seeks to realize this idea to a working device, capable of alerting a user wearing headphones to an external acoustic event.

## Project Objectives

The primary objective of the Group 9 Senior Design Project is to build a device known as the Acoustic Awareness Enabler, which is described as follows. The Acoustic Awareness Enabler is a device that is positioned between two auxiliary cables leading from an audio source to headphones. The device listens to outside noises and alerts the user (i.e. wearing headphones) to the unusually loud sound by turning off the audio throughput. The device uses three sensitivity settings, which provide the user with a selection of necessary threshold sound level (relative to surrounding environment) needed to pause the audio. The following were outlined as customer requirements for the Acoustic Awareness Enabler:

Table 1 Customer Requirements

|  |  |
| --- | --- |
| **Cust. Req. #** | **Customer Requirements Description​** |
| 1 | Self-containment of power source (rechargeable) and battery life of 8 hours​ |
| 2 | 3.5mm audio jack for input and output​ |
| 3 | Must have 3 different levels of interruption noise sensitivity (High, Medium and Low sensitivity) ​ |
| 4 | When interruption is detected, the volume of the noise going through users’ headphones must be muted or lowered​ |
| 5 | Must be able to distinguish between ambient and interrupt with a 1% false interrupt detection rate​ |
| 6 | Must have a reset button to restart audio ​ |
| 7​ | Separate audio cable between device and audio source​ |

## Concept of Operations

In terms of user interaction with the Acoustic Awareness Enabler, there arise only the following situations: turning the device on or off, changing the sensitivity threshold and resetting the device to start a new cycle of data collection or to turn the audio back on after it was turned off as a result of an interrupt.



Figure 1 Design Concept

Behind the scenes, the Acoustic Awareness Enabler is taking in acoustic data and testing whether a recent level exceeds a threshold level calculated from the average and standard deviation of a longer period of data. If the device finds that the sound level has not been exceeded, it will replace the oldest data set with the most recent and calculate the average and standard deviation continuously and test if a yet more recent sound will exceed a new threshold. If the threshold is exceeded, the device will turn off the audio throughput to the headphones. The Software Diagram flowchart outlines how the device works “under the hood”. In attempt to outline how the Acoustic Awareness Enabler works functionally, a matrix was constructed to outline what events would be triggered at which time under all possible conditions. Finally, a high-level schematic was designed to give an overview of what hardware would be used to build the Acoustic Awareness Enabler.

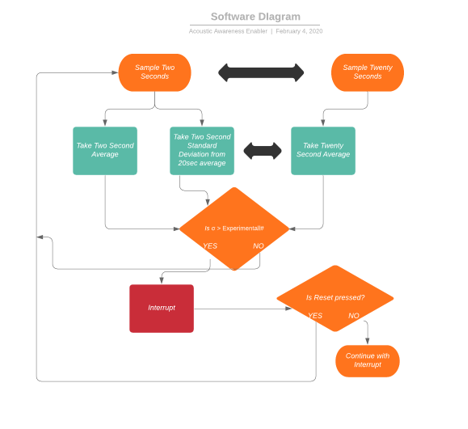


Figure 2 Software Diagram

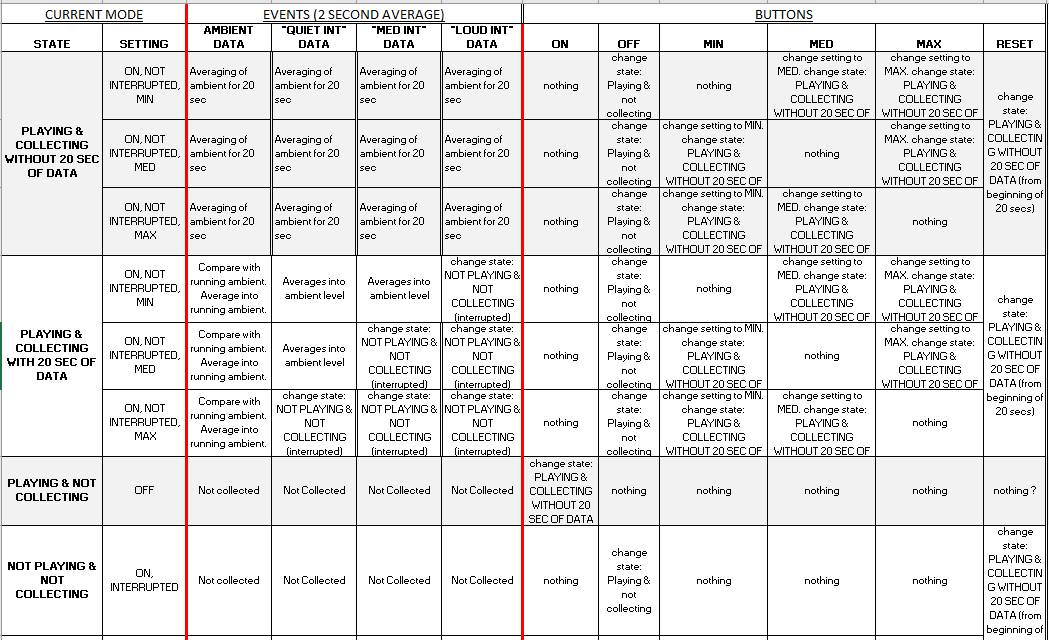
**

Figure 3 Concept of Design

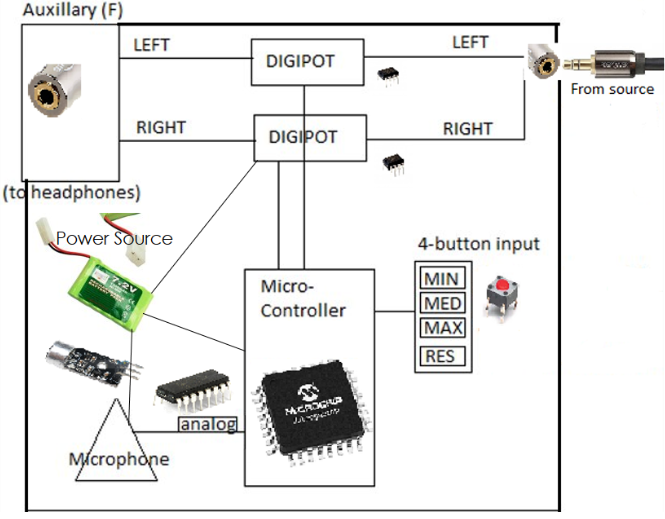


Figure 4 Device Example

# Engineering Requirements

## Verification Cross Reference Matrix (VCRM)

Presented below are the current engineering requirements. There has not been any significant change to the Engineering Requirements since the Preliminary Design Review (PDR), however some of the language has been updated to reflect consistency with respect to what has been completed already and to avoid confusion with current engineering decisions. A new column is added to reflect the related Test Plans associated with each Engineering Requirement and further, how Test Plans are used to verify an Engineering Requirement; either by inspection, analysis, demonstration, test or a combination of these verification methods. Where needed, further explanation is provided in the final column to clarify how the proposed Test Plans will prove satisfactory for each Engineering Requirement.

Table 2 Verification Cross Reference Matrix

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Cus.Req. #** | **Eng. Req. #** | **Engineering Requirement Description** | **Related Test Plan Codes** | **Verification** | | | | **Where met in design** |
| **I** | **A** | **D** | **T** |
| 1​ | 1.1 | 8 hours of power supplied to device​. | ET\_09, ET\_13, ET\_15 |  |  | X | X | Power supplied by battery. |
| 2​ | 2.1 | Audio Source Input 3.5mm ​ | ET\_04, NT\_02 | X |  | X |  | Interface of enclosure and electronics. |
| 2.2 | Audio Source Output 3.5mm​ | ET\_04, NT\_02 | X |  | X |  |
| 2.3 | I/O interface – digital switch​ | ET\_03, ET\_07, | X |  |  | X | Digital Switch found in electronics and controlled by µC. |
| 3​ | 3.1 | Processing includes average and standard deviation calculations to determine threshold | PT\_01, PT\_02, PT\_04 |  | X | X |  | Process first validated in external software then programmed in µC. |
| 3.2 | Three separate sensitivity buttons | ET\_05, PT\_01 |  | X |  | X | Interface of enclosure and electronics. |
| 4​ | 4.1 | Digital switches turn off audio throughput when interruption threshold is exceeded. | ET\_10, ET\_11, ET\_12 |  |  |  | X | µC reads acoustic signals and alerts when threshold exceeding, sending a signal to the digital switch. |
| 5​ | 5.1 | Digital signal processing is invoked to determine the presence of an outlier in a set of acoustic level readings. ​ | PT\_04, CT\_01, CT\_04, CT\_05 |  |  | X |  | Process first validated in external software then programmed in µC. |
| 5.2 | Lowest Ambient Noise of 30 dB based on ANSI standards | ET\_01, PT\_01 |  | X |  | X | Microphone meeting requirement is selected, signal processing design in accordance. |
| 5.3 | Must have 100 Hz Sampling frequency to detect averages in sound (not reconstructing signal) ​ | CT\_05 |  |  | X |  | At the interface of the ADC on µC and analog input from microphone and analog electronics. |
| 5.4 ​ | Need a sound level detector | ET\_01 |  |  |  | X | Microphone is used. |
| 6​ | 6.1 | Physical reset button will be present on Acoustic Awareness Enabler to reset acoustic environment data. ​ | ET\_05, NT\_02 |  |  | X | X | Button pressed sends signal to µC to turn switch on after being off. If switch is on, clear processing data. |
| 6.2 | Use of switches to resume audio after reset button has been pressed. ​ | ET\_10 |  |  |  | X | Button pressed sends signal to µC to turn switch on after being off. |
| 6.3 | Button will recalibrate device and will delay further interrupts for 20 seconds while gathering data​ | ET\_11, PT\_04 |  |  | X | X | If switch is on, clear processing data. Set 20 second pause on interrupts. |
| 7​ | 7.0 | Audio (auxiliary) Cable Included with Enclosure | ET\_04 | X |  |  |  | Include test auxiliary cable. |

# Test Plans Overview and Flowchart

The use of test plans offers immense benefits to a design project. For instance, test plans allow for greater ease in troubleshooting by allowing a team member to trace their steps back to resolve an issue. A project is made verifiable with test plans, which increases the trust of the customer. The process of building test plans, by forcing the team to review their future actions in detail, improve the accuracy at which a team can predict the amount of time required to complete the project, in addition to anticipating potential pitfalls before they appear.

The test plans for the Acoustic Awareness Enabler are organized into the following categories: electronics, processing, coding, printed circuit board and enclosure. Testing is needed for all these components of the project. By no means does the categorization of a particular test plan exclude it from interacting with other test categories, however since all test plans contribute to the final goal which is a blend of all categories.

## Test Plan Flow Chart

It was also decided that most test plans require a prerequisite test plan. To cleanly outline the order in which test plans need to be made, a flowchart has been constructed:

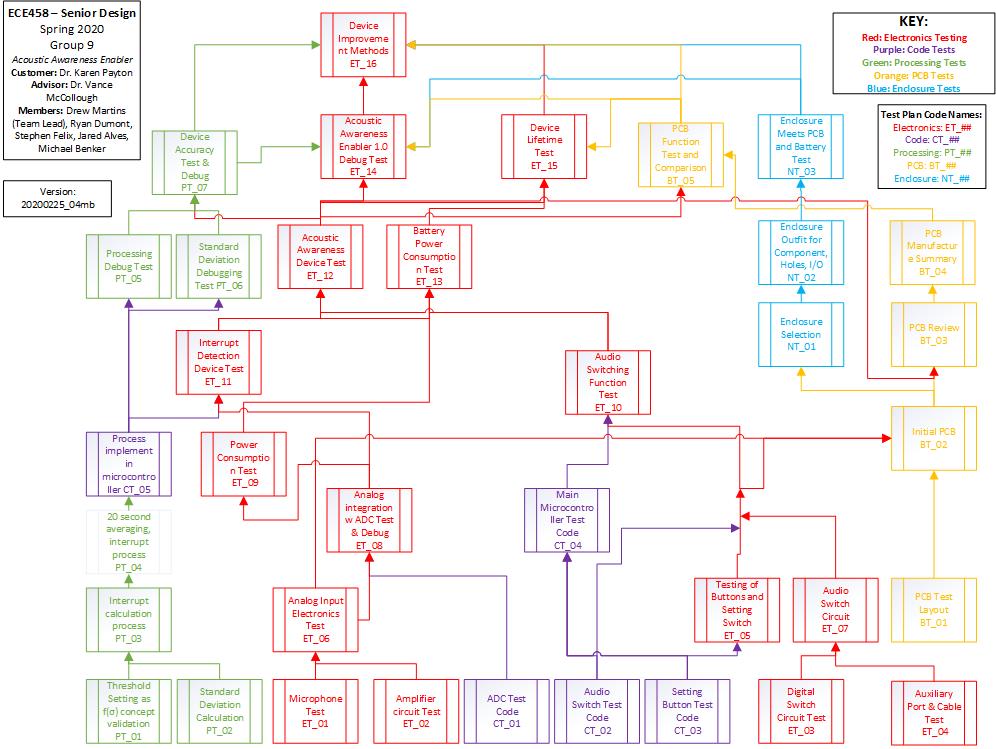


Figure Test Plan Flow Chart

## Test Plan Verification Matrix

A Verification Cross Reference Matrix is useful for identifying how Engineering Requirements are met using various Test Plans. A verification matrix may also be constructed as a high-level document outlining how each Test Plan proves useful to the completion of the project through satisfying an Engineering Requirement. It also provides an overview for all Test Plans and the proposed verification methods of each test.

Table 3 Test Plan Verification Matrix

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Code** | **Test Name** | **Verification Method** | | | | | **Prerequisite Test(s)** |
| **I** | **A** | **D** | **T** | **Comments** |  |
| ET\_01 | Microphone Test |  |  |  | T |  | None |
| ET\_02 | Amplifier Circuit Test |  |  |  | T |  | None |
| ET\_03 | Digital Switch Test |  |  |  | T |  | None |
| ET\_04 | Auxiliary Port and Cable Test | I |  |  |  |  | None |
| ET\_05 | Testing of Buttons and Switch |  |  |  | T |  | None |
| ET\_06 | Analog Input Electronics Test |  |  |  | T |  | ET\_01, ET\_02 |
| ET\_07 | Audio Switch Test | I |  |  |  |  | ET\_03, ET\_04 |
| ET\_08 | Analog Integration w ADC Test and Debug |  |  |  | T |  | ET\_06, CT\_01 |
| ET\_9 | Power Consumption Test |  |  |  | T |  | ET\_08 |
| ET\_10 | Audio Switching Function |  |  |  | T |  | ET\_07,CT\_04 |
| ET\_11 | Interrupt Detection Device |  |  |  | T |  | ET\_08, CT\_05 |
| ET\_12 | Acoustic Awareness Device |  |  |  | T |  | ET\_10, ET\_11 |
| ET\_13 | Battery Power Consumption |  |  |  | T |  | ET\_09, ET\_10, ET\_11 |
| ET\_14 | Acoustic Awareness Enabler 1.0 Debug Test |  |  |  | T |  | ET\_12, PT\_07, BT\_05, NT\_03 |
| ET\_15 | Device Lifetime Test |  |  |  | T |  | ET\_12, ET\_13, BT\_05 |
| ET\_16 | Device Improvement Methods | I |  |  |  |  | ET\_14, ET\_15, PT\_07, BT\_05, NT\_03 |
| PT\_01 | Threshold Setting as f(σ) Concept Validation |  | A |  |  |  | None |
| PT\_02 | Standard Deviation Calculation |  | A |  |  |  | None |
| PT\_03 | Interrupt Calculation Process |  | A |  |  |  | PT\_01, PT\_02 |
| PT\_04 | 20 Second Averaging, Interrupt Process |  |  | D |  |  | PT\_03 |
| PT\_05 | Processing Debug Test |  |  |  | T |  | CT\_05 |
| PT\_06 | Standard Deviation Debugging |  |  |  | T |  | ET\_11 |
| PT\_07 | Device Accuracy Test & Debug |  |  |  | T |  | ET\_12, PT\_05, PT\_06 |
| CT\_01 | ADC Test Code |  |  |  | T |  | None |
| CT\_02 | Audio Switch Test Code |  |  |  | T |  | None |
| CT\_03 | Setting Button Test Code |  |  |  | T |  | CT\_01 |
| CT\_04 | Main µC Test Code |  |  |  | T |  | CT\_02, CT\_03 |
| CT\_05 | Process Implement in µC |  |  | D |  |  | PT\_04 |
| BT\_01 | PCB Test Layout |  |  | D |  |  | None |
| BT\_02 | Initial PCB |  |  | D |  |  | ET\_06, BT\_01 |
| BT\_03 | PCB Review | I |  |  |  |  | ET\_12, BT\_02 |
| BT\_04 | PCB Manufacture Summary | I |  |  |  |  | BT\_03 |
| BT\_05 | PCB Function Test and Comparison | I |  |  |  |  | ET\_12, BT\_04 |
| NT\_01 | Enclosure Selection | I |  |  |  |  | ET\_09, BT\_02 |
| NT\_02 | Enclosure Outfit for Components, I/O |  |  | D |  |  | NT\_01 |
| NT\_03 | Enclosure Integration with Battery and PCB | I |  |  |  |  | BT\_04, NT\_02 |

# Test Plans: Electronics

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Name** | Microphone Test | | **Test Number** | ET\_01 | | | |
| **Requirement(s) Tested** | | 5.4, 5.2 | **Verification Method** | I | A | D | T |
| **Test Setup** | | Fig. 6  Oscilloscope  Variable DC Power Supply  Microphone Part Number: PM0F-6050P-36UQ  website: noises.online | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Step** | **Action (Attach test data, diagrams, etc. as appropriate)** | **Expected Result** | **Observed Result** | **Pass**  **Fail** |
| 1 | Build Microphone Circuit as shown in Fig. 6 | See Fig. 6 |  |  |
| 2 | Turn on and set power supply to 5VDC and connect to Vcc on circuit diagram; connect ground. | 5VDC on Vcc, verifiable with volt meter |  |  |
| 3 | Turn on oscilloscope, probe output of microphone and connect your ground wire. | Voltage output between 10 and 500 mV |  |  |
| 4 | Select noise, (website: noises.online) play sound using speakers and record output in dB with sound meter. Ensure background noises are minimal. | dB level between 30 and 90 dB; appropriate for volume level |  |  |
| 5 | Record voltage output from microphone. | Between 10 and 500 mV |  |  |
| 6 | Repeat steps 4 and 5 for different volume levels with the same sound selection. | Voltage output should be between 10 and 500 mV and voltage output increases with volume level increase. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Name** | Amplifier Circuit Test | | **Test Number** | ET\_02 | | | |
| **Requirement(s) Tested** | | 5.4 | **Verification Method** | I | A | D | T |
| **Test Setup** | | See Figure 7  Variable DC Power Supply  Oscilloscope | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Step** | **Action (Attach test data, diagrams, etc. as appropriate)** | **Expected Result** | **Observed Result** | **Pass**  **Fail** |
| 1 | Build Amplifier Circuit as shown in Fig. 7 | See Fig. 7 |  |  |
| 2 | Turn on Power Supply and connect 5V DC to Amplifier supply rails (see Fig 7). | Voltage output verifiable with Volt Meter |  |  |
| 3 | Turn on Function Generator and select (any) waveform with voltage range 0-500mV. Turn on Oscilloscope and connect probes to function generator output. | Oscilloscope window displays waveform. |  |  |
| 4 | Connect function generator output to input of amplifier circuit (see Fig. 7). Probe Channel 1 oscilloscope probes to amplifier input. Connect oscilloscope Channel 2 probes to amplifier output (see Fig 7). | Expected gain ~11. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Digital Circuit Switch Test | | Test Number | ET\_03 | | | |
| Requirement(s) Tested | | 4.1 | Verification Method | I | A | D | T |
| Test Setup | | Figure 9  Digital Switch Part Number: CD4066BE  DC Power Supply (For Switch’s CTRL input and power source)  Function Generator (For the input signal, an AC power source)  Oscilloscope (Measuring the input and output signal of the Switch) | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Build Digital Switch Circuit as shown in Figure 9 | See Figure 9 |  |  |
| 2 | Turn on power Supply and connect 5V DC to the VDD of the Digital Switch Circuit. | Voltage output verifiable with Volt Meter |  |  |
| 3 | Use the Function Generator to send out a 5V AC Sine Signal and then test with a 5V AC Noise Signal. The signal will go to the input of one of the Digital Switches on the chip. | Voltage output verifiable with oscilloscope. |  |  |
| 4 | Set the CTRL pins of a switch between 5V and GND. Then compare the results on Step 5. | Voltage output verifiable with Volt Meter. |  |  |
| 5 | Connect an Oscilloscope (1x) Probe to the input signal (Vin) and output signal (Vout) and compare the two signals. Connect the grounds as well. | Verifiable through Oscilloscope:  If CTRL = 5V  Vout = Vin  If CTRL = GND  Vout = 0V DC |  |  |

|  |
| --- |
| Comments: |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Auxiliary Port & Cable Test | | Test Number | ET\_04 | | | |
| Requirement(s) Tested | | 2.1,2.2,5.1,6.3,7.0 | Verification Method | I | A | D | T |
| Test Setup | | Audio Source (computer or phone)  3.5mm headphone jack(x2)  Male-to-Male Auxiliary Cable  Headphones | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1. | Attach the 3.5mm headphone jack to the input and output to make sure it fits. | The Auxiliary jack should fit any standard 3.5mm audio device. |  |  |
| 2 | Connect Auxiliary ports together in series. |  |  |  |
| 3 | Connect the M-M auxiliary cable from an audio source to the auxiliary ports and connect headphones to the output of the other auxiliary port. Observe audio output to headphones. | The audio signal would travel through the device with little delay and drop of signal. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Testing of Buttons and Setting Switch | | Test Number | ET\_05 | | | |
| Requirement(s) Tested | | 3.2, 6.1 | Verification Method | I | A | D | T |
| Test Setup | | Signal Detector (Multimeter or Oscilloscope)  Voltage Signal | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Connect one pin of each button to a voltage signal (5VDC). Measure input and output voltage with multimeter. | When Button is pressed the open circuit should become closed and the voltage signal should go through. |  |  |
| 2 | Connect the Switch to a voltage signal and ground. Measure input and output voltage with multimeter. | Just like the buttons, when the switch is pressed check if current flows. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Name** | Analog Input Electronics Test | | **Test Number** | ET\_06 | | | |
| **Requirement(s) Tested** | | 5.4 | **Verification Method** | I | A | D | T |
| **Test Setup** | | Fig. 6 7, 8  Sound meter Voltmeter  Variable DC Power Supply  Oscilloscope  website: noises.online | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test Step** | **Action (Attach test data, diagrams, etc. as appropriate)** | **Expected Result** | **Observed Result** | **Pass**  **Fail** |
| 1 | Verify Tests ET\_01, ET\_02. Build Microphone Test Circuit and Amplifier Test Circuit (see Fig. 6 and 7) | See Fig. 6 and 7 |  |  |
| 2 | Connect output of Microphone Circuit to input of Amplifier Circuit. | See Fig. 8 |  |  |
| 3 | Turn on Power Supply and connect 5VDC to Amplifier supply rails (see Fig 8). | Voltage output verifiable with Volt Meter |  |  |
| 4 | Turn on Oscilloscope and connect Channel 1 probe to Microphone output (e.g. Amplifier Input) and Channel 2 probe to amplifier output. | Observable amplification (A=11) of ambient. |  |  |
| 5 | Select noise, (website: noises.online) play sound using speakers and record output in dB with sound meter. Ensure background noises are minimal. | dB level between 30 and 90 dB; appropriate for volume level |  |  |
| 6 | Record voltage output from microphone and from amplifier. | Expected gain ~11 |  |  |
| 7 | Repeat steps 4 and 5 for different volume levels with the same sound selection. | Voltage increases linearly with sound level reading (dB). |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Audio Switch Circuit | | Test Number | ET\_07 | | | |
| Requirement(s) Tested | | 2.1, 2.2, 2.3, 4.1 | Verification Method | I | A | D | T |
| Test Setup | | Fig. 9  Digital Switch Part Number: CD4066BE)  DC Power Supply (For Switch’s CTRL input and power source)  A Male-to-Male Auxiliary Cable from a device to the circuit & a pair of headphones or speakers.  Two Auxiliary Ports (One Aux-Input and One Aux-Output) | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Verify tests ET\_03 and ET\_04 | See ET\_03 and ET\_04 |  |  |
| 2 | Connect a 5V DC power source to the VDD on the Switches | Voltage output verifiable with Voltmeter |  |  |
| 3 | Set up two 5V DC inputs for the CTRL pins, one for each switch | Voltage output verifiable with Voltmeter |  |  |
| 4 | Connect an audio source (auxiliary cable – male-male) to the Aux-Input port | Auxiliary cable inserted |  |  |
| 5 | Connect headphones or speaker to Aux-Output and begin any mp3 file. | Audio can be heard if switch is on. |  |  |
| 6 | Set the CTRL pins of each switch between 5V and GND and determine if the audio is getting through to the other side, | If CTRL = 5V  Audio is enabled  If CTRL = GND  Audio is muted |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Analog Integration with ADC Test & Debug | | Test Number | ET\_08 | | | |
| Requirement(s) Tested | | 3.1, 5.1, 5.2, 5.3, 5.4 | Verification Method | I | A | D | T |
| Test Setup | | Integration of ET\_06 and CT\_01  ATmega328PB  Variable DC Power Supply  Atmel Studio | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Verify functionality of ET\_06. | Complete ET\_06 |  |  |
| 2 | Connect the output of the op- amp to pin C5 on the microcontroller. | ET\_06 will be able to communicate with microcontrollers’ ADC. |  |  |
| 3 | Run CT\_01. | See CT\_01 |  |  |
| 4 | Open the Atmel Studio software on your computer and use the debug function. |  |  |  |
| 5 | View the values that are seen by the ADC as ET\_06 is taking in the ambient noise. |  |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Power Consumption Test | | Test Number | ET\_09 | | | |
| Requirement(s) Tested | | 1.1 | Verification Method | I | A | D | T |
| Test Setup | | Integration of ET\_08 and ET\_05  ATmega328PB  Power Supply  Digital Multimeter | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Build ET\_08 and load CT\_05 in microcontroller. | Program successfully loaded. See ET\_08. |  |  |
| 2 | Set power supply to 5 Vdc and connect the positive lead to Vcc and negative lead to ground. | Verifiable with voltage meter |  |  |
| 3 | Take digital multimeter and set it to read current. | Multimeter should switch to amperage mode. |  |  |
| 4 | Disconnect positive lead of power supply and connect one end of meter to that wire and the other lead of meter to Vcc on the circuit. | This will complete the circuit and allow you to measure current draw. |  |  |
| 5 | Run circuit for an hour and log the amperage output on multimeter for five-minute intervals | Current values should be roughly in the same range of each other. |  |  |
| 6 | Take average of those numbers and that is roughly the current draw of circuit per hour. Multiply this number times 8 (# of hours device needs to run for) and record the number. This is the current rating you will need on your battery. | An average of the current levels will be obtained to help with battery selection. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Audio Switching Function Test | | Test Number | ET\_10 | | | |
| Requirement(s) Tested | | 3.2 | Verification Method | I | A | D | T |
| Test Setup | | Variable DC Power Supply  ATmega328PB  Integration of CT\_04 and ET\_08 | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Have ET\_08 already built and CT\_04 running on the microcontroller. | See ET\_08 and CT\_04. |  |  |
| 2 | Implement the code from CT\_04 with the circuit, testing the switch with the various sensitivity levels | Switch circuit functions with sensitivity levels accurately |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Interrupt Detection Device Test | | Test Number | ET\_11 | | | |
| Requirement(s) Tested | | 3.1, 4.1, 5.1, 5.4 | Verification Method | I | A | D | T |
| Test Setup | | Integration of ET\_07 and CT\_04  ATmega328PB  Variable Power Supply  Digital Multimeter  Oscilloscope | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Make sure ET\_07 is built and functioning correctly and CT\_04 is running on the microcontroller. | See ET\_07 and CT\_04. |  |  |
| 2 | Power up the variable DC power supply and set the output voltage to 5 VDC. | Verifiable with voltmeter. |  |  |
| 3 | Connect the positive lead to the +5 VDC voltage rail and the negative lead to the ground rail of the circuit. | Circuit should have power at this point. |  |  |
| 4 | Power on oscilloscope and connect test probes to channel one and channel 2. | Oscilloscope will be ready to view ambient noise. |  |  |
| 5 | Probe the input and output with channel 1 and channel 2 respectively to make sure microphone and amplifier are reading the ambient noise in the environment. | Original and amplified signal should be in view on screen. |  |  |
| 6 | Create a loud noise of your choosing for about 2 seconds to trigger an interrupt. | Loud noise audible. |  |  |
| 7 | Verify that an interrupt is achieved, examining the output pin of the microphone designated for the interrupt. | Interrupt is achieved. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Acoustic Awareness Device Test | | Test Number | ET\_12 | | | |
| Requirement(s) Tested | |  | Verification Method | I | A | D | T |
| Test Setup | | Integration of ET\_09, ET\_10 and ET\_11  Website: noises.online  Sound Meter | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Complete Tests ET\_11 and ET\_12. | Verify ET\_11 and ET\_12 are functioning. |  |  |
| 2 | Connect the interrupt output pin on the microcontroller from ET\_12 to the input of ET\_11. Monitor the output pin using a voltmeter or oscilloscope probe. | Follow circuit diagram for ET\_13. |  |  |
| 3 | Begin audio throughput through the auxiliary cables from the device source to the switching circuit and out to the headphones. | Sound coming from device source is on. |  |  |
| 4 | Select an ambient noise level from website: noises.online as was done in ET\_06. Allow device to maintain a constant ambient reading for 20 seconds. | Sound selection is audible for 20 seconds. |  |  |
| 5 | After 20 seconds, rapidly increase the volume of the output of the sound to start an interrupt. Monitor the audio output to the headphones. | Audio to the headphones should turn off when an interrupt is started. Ambient sound selection is much louder than the previous step to trigger and interrupt. |  |  |
| 6 | Repeat step 5 and record ambient levels before and after the interrupt while taking record of whether the audio was turned off at the time of the ambient volume increase. | Audio should turn off at volume increase IF the threshold has been exceeded. |  |  |
| 7 | Repeat steps 5 and 6 for all threshold level settings (high, medium, low). | Record under which cases the audio was turned off and the setting, ambient and interrupting sound level. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Battery Power Consumption Test | | Test Number | ET\_13 | | | |
| Requirement(s) Tested | | 1.1 | Verification Method | I | A | D | T |
| Test Setup | | Integration of ET\_11 and ET\_09  ATmega328PB  Selected Lithium Ion Battery  Stopwatch | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Using calculated current consumption in ET\_09, choose a lithium ion battery pack from an online retailer that has at least that much or more capacity. | Battery has enough capacity for 8 hours. |  |  |
| 2 | Connect battery to ET\_11 and run CT\_05 for 8 hours. | The device should run on battery power for at least 8 hours. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Name | Acoustic Awareness Enabler 1.0 Debug Test | Test Number | ET\_14 | | | |
| Requirement(s) Tested | All | Verification Method | I | A | D | T |
| Test Setup | Integration of BT\_05, ET\_12, PT\_07 and NT\_03 | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Ensure that all components are housed and properly mounted in the modified enclosure. | A “completed” device will be built. |  |  |
| 2 | Turn on the Acoustic Awareness Enabler. | Device powers up. |  |  |
| 3 | Use one aux cord to connect your computer to the input jack of the A.A.E | This is where the music input will come from |  |  |
| 4 | Connect your headphones from the output of the A.A.E to your ears. | This will allow you to hear the audio signal. |  |  |
| 5 | Test each sensitivity level individually by pressing the panel mounted button associated with it. | Threshold setting changes to desired threshold setting. |  |  |
| 6 | Create various “interrupts” for each sensitivity level test. | Sound traveling through headphones should stop. Depending how well it stops may mean software adjustments for thresholds. |  |  |
| 7 | Log the behavior of the device and see if fine tuning in software is needed. | Results will tell you what work still needs to be done. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Device Lifetime Test | | Test Number | ET\_15 | | | |
| Requirement(s) Tested | | 1.1 | Verification Method | I | A | D | T |
| Test Setup | | Completed Circuit Board  Chosen Battery Pack | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Verify that BT\_05 is satisfied | Circuit board should be populated and ready to test. |  |  |
| 2 | Take chosen battery and wire the positive terminal to Vcc on the final circuit board and the negative terminal to ground. Keep battery attached for 8 hours and turn on circuit. Do various tests during the 8-hour interval. | Battery should not die for the full length of time. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Device Improvement Methods | | Test Number | ET\_16 | | | |
| Requirement(s) Tested | | All | Verification Method | I | A | D | T |
| Test Setup | |  | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Confirm NT\_03, BT\_05, ET\_15, ET\_14, PT\_07 | See NT\_03, BT\_05, ET\_15, ET\_14, PT\_07. |  |  |
| 2 | Verify each and every category for possible improvements: Enclosure, PCB and electronics | Final product |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

# Test Plans: Processing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Threshold Setting as f(σ) concept validation | | Test Number | PT\_01 | | | |
| Requirement(s) Tested | | 5.1 | Verification Method | I | A | D | T |
| Test Setup | | MATLAB  In conjunction with PT\_02 | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Obtain data set from environment X with sound meter, collecting 30 samples from “quiet environment”. Collect ambient samples, simulate “Low”, “Med” and “High” Interrupts and check sound levels | Ambient levels (dB) and sound levels of low, medium and high interrupts |  |  |
| 2 | Repeat step 1 for “Medium” environment and “Loud” Environment | New data sets with same value categories |  |  |
| 3 | Compare and compute standard deviations in MATLAB of various data sets | Standard deviations of data sets |  |  |
| 4 | Verify that threshold from experimental data is based on standard deviation | Threshold Setting as a function of the standard deviation |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Standard Deviation Calculation | | Test Number | PT\_02 | | | |
| Requirement(s) Tested | | 5.1,5.2 | Verification Method | I | A | D | T |
| Test Setup | | MATLAB  In conjunction with PT\_01 | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Generate a plot of ambient level vs standard deviation in MATLAB by comparing measured data and standard deviations for data sets | Determine the relationship between the standard deviation and the ambient level of the environment |  |  |
| 2 | Derive a formula for threshold levels based on ambient and standard deviation | Find relationship between threshold setting and standard deviation |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Interrupt Calculation Process | | Test Number | PT\_03 | | | |
| Requirement(s) Tested | | 5.1,5.2,5.3 | Verification Method | I | A | D | T |
| Test Setup | | MATLAB – “.wav Analyzer” and “Random Signal Analyzer” | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Use derived formula from PT\_02 | From PT\_02 |  |  |
| 2 | Use data of ambient levels and interruption levels for each threshold setting to see how many standard deviations away the interrupts are | Interruption calculation |  |  |
| 3 | Use .wav Analyzer and Random Signal Analyzer to compare random data thresholds and .wav thresholds to calculated thresholds for validation | Validation of interruption calculation |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | 20 second averaging, interrupt process | | Test Number | PT\_04 | | | |
| Requirement(s) Tested | | 5.1,5.2,5.3 | Verification Method | I | A | D | T |
| Test Setup | | Microsoft Visual Studio, C#  See attachment Code (Interrupt.cs) | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Write and compile code in C# that simulates the collection of 20 seconds or more of data at a rate of 100 samples per second. | Stores 20 seconds of data. |  |  |
| 2 | Implement standard deviation calculation to determine if a 2 second interval after 20 seconds of data is within a number of standard deviations. If the most recent 2 seconds is above a number of standard deviations, the program should indicate it. | Standard deviation calculation of interrupt is verifiable through calculation. |  |  |
| 3 | Optimize the code RAM space using averaging so that only one data set is saved for every two second interval. | Verify that only 10 numbers are saved and are used in the average and standard deviation calculation. |  |  |
| 4 | If not present in the test code thus far, modify the code so that the oldest number in the data set is replaced by the newest average if the newest average is within a number of standard deviations. | After 20 seconds, data stored should always be from the most recent 20 seconds if there were no interrupts. See attachment code for final example of implementation. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Process Debug Test | | Test Number | PT\_05 | | | |
| Requirement(s) Tested | | 5.1 | Verification Method | I | A | D | T |
| Test Setup | | Microsoft Visual Studio, C# | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Test different averaging times (25 second average vs 3 second average) within C# by updating program | Data on 25 second average and 3 second average |  |  |
| 2 | Implement recursive averaging in software (C#) | Comparison of recursion to multi-averaging method |  |  |
| 3 | Test averaging to see if a time delay occurs when taking the average using Atmel studio due to instruction timing | Time delay confirmation |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Standard Deviation Debugging | | Test Number | PT\_06 | | | |
| Requirement(s) Tested | | 5.1 | Verification Method | I | A | D | T |
| Test Setup | | Microsoft Visual Studio, C# | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | After ET\_11, see if number of standard deviations must be fine tuned | Result of ET\_11 |  |  |
| 2 | Test various standard deviation multiples close to the values originally chosen to see if different multiples of the standard deviation give more accurate threshold | Fine-tuned standard deviation multiples |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Device Accuracy Test & Debug | | Test Number | PT\_07 | | | |
| Requirement(s) Tested | | 5.1,5.4 | Verification Method | I | A | D | T |
| Test Setup | | Integration of BT\_05, PT\_05 and PT\_06 | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Confirm that Customer Requirement 5 is met by checking that the device is accurate 99% of the time through repetitive testing of device | Requirement 5 |  |  |
| 2 | Repeat testing of device as many times until 99% accuracy is achieved, fine tuning PT\_05 and PT\_06 as necessary | 99% efficiency and Customer Requirement 5 met |  |  |
| 3 | Change to new environment with a wider variance of ambient values and repeat process | Accurate device in varying environments |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

# Test Plans: Coding

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | ADC Test Code | | Test Number | CT\_01 | | | |
| Requirement(s) Tested | | 5.1 | Verification Method | I | A | D | T |
| Test Setup | | Microsoft Visual Studio, C  See Appendix Code(ADC Test Code) | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Begin Microsoft Visual Studio and select C as language | N/A |  |  |
| 2 | Initialize the I/O and interrupt header files | Allows for I/O and interrupt libraries to be used |  |  |
| 3 | Write a function to clear two ports (ex. B and D) | Function is usable in main |  |  |
| 4 | Within main function, set Data Direction registers B, C and D as necessary | Data Direction Registers configured |  |  |
| 5 | Initialize ADMUX and ADC Status and Control Registers (reserved voltage and ADC channel) | Correct ADC channel and voltage configuration |  |  |
| 6 | Enable global interrupts and set timer1 using TCNT1 and TMSK1 as well as control registers | Interrupts enabled to be used for step 7 |  |  |
| 7 | Write Interrupt Service Routine to display highest two digits and lowest 8 digits of ADC | Tested ADC |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Name | Audio Switch Test Code | Test Number | CT\_02 | | | |
| Requirement(s) Tested | 2.3, 4.1, 6.2 | Verification Method | I | A | D | T |
| Test Setup | Processor: ATmega328PB  Program: Test Code for Buttons  References: ATmega328PB Datasheet  Hardware: CD4066BE Quad Bilateral Switch (14-DIP) Chip | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Write C code that tests the CPU and Switches |  |  |  |
| 2 | Set PINB1:0 as outputs to each Switch (PB1 for Aux-Left and PB0 for Aux-Right). | PINB1 -> Aux Left  PINB0 -> Aux Right |  |  |
| 3 | Run a loop that counts PINB1:0 from 0 to 3, increment by 1 per second. | t = 0s, PB1:0 = 0  t = 1s, PB1:0 = 1  t = 2s, PB1:0 = 2  t = 3s, PB1:0 = 3  t = 4s, PB1:0 = 0 |  |  |
| 4 | Test Circuit and Code | The sound should turn on each side of Aux depending on the state of PB1 and PB0. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | | Setting Button Test Code | | Test Number | CT\_03 | | | |
| Requirement(s) Tested | | 3.2, 6.1, 6.3 | Verification Method | I | A | D | T |
| Test Setup | | | Processor: ATmega328PB  Program: Test Code for Buttons  References: ATmega328PB Datasheet  Hardware:   * 5x Push Buttons w/4.7kΩ Pull-up Resistor. * 3x LED lights w/330Ω Resistor. | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Write C code that tests the CPU and Buttons |  |  |  |
| 2 | Set PIND2:0 as output for the LED, each LED represents different threshold levels | PD2:0 is the output |  |  |
| 3 | Initialize PINC4:0 as the pull-up input for each button. | PINC4:0 are now pull-up inputs for the buttons to use |  |  |
| 4 | Implement button PINC3 as the RESET button. | PINC3 restarts the device. |  |  |
| 5 | Implement button PINC2:0  as Threshold Sensitivity buttons. Sets the Sensitivity levels. | PINC2 = High  PINC1 = Medium  PINC0 = Low |  |  |
| 6 | Implement button PINC4 to set the device to sleep mode and restart the program if the button is pressed again. | Sleeps on 1 button press and restarts the device on the next press. |  |  |
| 7 | Test Circuit and Code | PC4 Button turns the power on/off  PC3 resets the program  PC2:0 changes threshold levels |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Main Microcontroller Test Code | | Test Number | CT\_04 | | | |
| Requirement(s) Tested | | 2.3, 3.2, 4.1, 6.1, 6.2, 6.3 | Verification Method | I | A | D | T |
| Test Setup | | Processor: ATmega328PB  Program: Integration of CT\_02 and CT\_03  References: ATmega328PB Datasheet | | | | |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Verify Tests CT\_02 and CT\_03 | See Tests CT\_02 and CT\_03 |  |  |
| 2 | Initialize PIND2:0 as outputs (for the LED lights) | PIND2:0 are outputs |  |  |
| 3 | Initialize PINC4:0 as the pull-up input for each button | PINC4:0 are now pull-up inputs for the buttons to use |  |  |
| 4 | Initialize PINB1:0 as inputs (for the Bilateral Switches) | PINB1 and PINB0 controls each switch |  |  |
| 5 | Run a loop that counts PINB1:0 from 0 to 3, increment by 1 per second. | t = 0s, PB1:0 = 0  t = 1s, PB1:0 = 1  t = 2s, PB1:0 = 2  t = 3s, PB1:0 = 3  t = 4s, PB1:0 = 0 |  |  |
| 6 | Test Circuit and Code (Switches) | The sound should turn on each side of Aux depending on the state of PB1 and PB0. |  |  |
| 7 | Test Circuit and Code (Buttons) | PC4 Button turns the power on/off  PC3 resets the program  PC2:0 changes threshold levels |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | | Process Implement in Microcontroller | | Test Number | | | CT\_05 | | |
| Requirement(s) Tested | | 6.2, 6.3 | | Verification Method | | I | A | | D | T |
| Test Setup | | Atmel Studio, Microsoft Visual Studio, AVR programming, C#, Oscilloscope and probes | | | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Perform demonstration PT\_04. | See PT\_04 Test Plan. |  |  |
| 2 | To implement process PT\_04 in microcontroller, match variables that are defined using I/O such as analog input and threshold setting to microcontroller memory. | All variables from PT\_04 should be accounted for, including cascaded averages, 20 second standard deviations. |  |  |
| 3 | Without connecting any I/O such as microphone and analog circuitry, digital switch and threshold switches, use the microcontroller to step through the process using user-defined 2 second ambient level averages. | Monitor 2 second averages saved to microcontroller. |  |  |
| 4 | Verify that after 20 seconds, the oldest 2-second average gets replaced by the newest if the newest is below the interruption threshold. | Verification complete. |  |  |
| 5 | Verify that after 20 seconds, if the newest 2-second average is above the threshold standard deviation, averaging stops and a high signal voltage is output to the digital switch output pin (not connected) using oscilloscope and probe. | Verification complete. |  |  |
| 6 | Verify that the RESET variable resets the 20 seconds of averaging data when activated and no interrupt is allowed for 20 more seconds. | Verification complete. |  |  |
| 7 | Verify that the OFF switch will tell the microcontroller to manually shut down, as opposed to cutting the voltage supply. | Verification complete. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

# Test Plans: Printed Circuit Board

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | PCB Test Layout | | Test Number | BT\_01 | | | |
| Requirement(s) Tested | | Constraint of maximum size: 3.5” x 2” | Verification Method | I | A | D | T |
| Test Setup | | Cadence OrCAD Suite  Eagle Autodesk PCB Designer Tools | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Run Cadence OrCAD: Capture CIS and construct a circuit with estimated components necessary for the PCB, including Microcontroller, Digital Switch, approximate analog electronics selections, OP-AMP and connections. | Circuit Layout |  |  |
| 2 | Import necessary library and padstack files (website: ultralibrarian.com) for components that are not standard in the program. | All components accounted for. |  |  |
| 3 | Export the schematic layout to a netlist file and load the netlist into OrCAD PCB Editor with a board layout size at the maximum constraint size 3.5” x 2” | Netlist file generated without errors or warnings. Board size in the Editor is 3.5” x 2”. Verify that components can be placed using the netlist. |  |  |
| 4 | Place components on the board in the software and connect wires to demonstrate that all components fit in the maximum constraint size. | All components fit on the 3.5” x 2” layout. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Initial PCB | | Test Number | BT\_02 | | | |
| Requirement(s) Tested | | 6.1, 6.2, 6.3, Constraint of maximum size: 3.5” x 2” | Verification Method | I | A | D | T |
| Test Setup | | Cadence OrCAD Suite  Eagle Autodesk PCB Designer Tools  Serpac Enclosures Website | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Verify Exact component selections and circuit layouts in ET\_06 and ET\_08. | See ET\_06 and ET\_08 |  |  |
| 2 | Create new schematic or modify layout from BT\_01 to include final component selections. Verify that all components and wire paths are correct. | Schematic components and wire pathways are correct. |  |  |
| 3 | Run Cadence OrCAD: Capture CIS or Eagle Autodesk PCB designer tools and construct a circuit with final components choices and correct wire pathways. | Circuit layout is in accordance with schematic. |  |  |
| 4 | If using Capture CIS, import necessary library and padstack files (website: ultralibrarian.com) for components that are not standard in the program. If using Eagle, ensure that all layout components are available. | All components accounted for. |  |  |
| 5 | Export the schematic layout to a netlist file and load the netlist into OrCAD PCB Editor or import the schematic to Eagle PCB Designer and ensure all layout components are available. | Netlist file generated without errors or warnings or all layout components available. |  |  |
| 6 | Reduce size of board as much as possible while preserving circuit layout on the board. | Board size is reduced from maximum constraint size. |  |  |
| 7 | Increase the board size to fit snug in the next largest available enclosure from Serpac enclosures website. Maintain equal margin sizes at each edge of the board. | Layout size fits in chosen enclosure. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | PCB Review | | Test Number | BT\_03 | | | |
| Requirement(s) Tested | | 6.1, 6.2, 6.3, Constraint of maximum size: 3.5” x 2” | Verification Method | I | A | D | T |
| Test Setup | | Cadence OrCAD Suite  Eagle Autodesk PCB Designer Tools | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Complete BT\_02 Initial PCB | See BT\_02 |  |  |
| 2 | Review component choices and pathways on PCB layout with teammates. | Component choices and pathways match schematic layouts. |  |  |
| 3 | Seek brief consultation with University sources including students, technicians and professors to confirm that PCB is acceptable and that parameters such as linewidth are appropriate for manufacture. | PCB is satisfactory or adjustments must be conducted. |  |  |
| 4 | If PCB design is less than satisfactory, adjust as necessary according to advice sought for PCB. | PCB is satisfactory for manufacture. |  |  |
| 5 | Seek methods of PCB manufacture and place order. | PCB ordered for manufacture. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | PCB Manufacture Summary | | Test Number | BT\_04 | | | |
| Requirement(s) Tested | | 7.0, constraint size. | Verification Method | I | A | D | T |
| Test Setup | | PCB(s) | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Complete BT\_03, send PCB out for manufacture and receive manufactured PCB(s). | PCB(s) is mailed out and in possession of the team. |  |  |
| 2 | Inspect all components on PCB to verify that the components are as expected. | Components are as expected. |  |  |
| 3 | Inspect all pathways on PCB to verify that they are as expected. | Pathways as expected. |  |  |
| 4 | Inspect all input and outputs paths on the PCB. | Inputs and outputs as expected. |  |  |
| 5 | Verify that PCB fits in selected enclosure. | PCB fits in enclosure with selected battery source(s). |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | PCB Function Test and Comparison | | Test Number | BT\_05 | | | |
| Requirement(s) Tested | | 6.1, 6.2, 6.3 | Verification Method | I | A | D | T |
| Test Setup | | PCB  Power Supply  Oscilloscope and probes  Website: noises.online  Sound meter | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Complete tests ET\_12 and BT\_04. | See ET\_12 and BT\_04. |  |  |
| 2 | Install program code to the ATmega328PB on the PCB. | ATmega328PB on PCB is responsive and code is loaded. |  |  |
| 3 | Connect wires to the connection ports on the PCB for each connector and connect power supply, ground and probes to respective ports on the PCB as per schematics used in ET\_12. | PCB function is comparable to ET\_12 function. |  |  |
| 4 | Conduct test ET\_12 but for PCB and record experimental results. | PCB function is comparable to ET\_12 function. |  |  |
| 5 | Compare results of PCB function test with ET\_12. | PCB function is comparable to ET\_12 function. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

# Test Plans: Enclosure

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Enclosure Selection | | Test Number | NT\_01 | | | |
| Requirement(s) Tested | | 7.0, Customer Req. 1 | Verification Method | I | A | D | T |
| Test Setup | | OrCAD Design  Battery/Circuit Board Dimensions | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Verify Battery Selection following battery consumption test ET\_09. | Battery is chosen and ready. |  |  |
| 2 | Verify size of PCB initial layout | Size should be within customer’s constraints. |  |  |
| 3 | Once length and width measurements of PCB are verified, pick an enclosure with correct length and width to house the board. | Length/Width of PCB and battery should be smaller than enclosure. |  |  |
| 4 | Make sure enclosure has enough height space within to allow battery and PCB to be placed on top of one another. | Enclosure should be high enough to hold battery and circuit board when stacked on top of each other. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Enclosure Outfit for Component, Holes, I/O | | Test Number | NT\_02 | | | |
| Requirement(s) Tested | | 1.1 | Verification Method | I | A | D | T |
| Test Setup | | Drill  Various Drill Bit Sizes  Enclosure  All Switches and Exterior Components | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Refer to figure 10 for tentative component placement. | A visual of proper placement will be obtained. |  |  |
| 2 | Drill the Proper size holes for each component in their appropriate place designated in figure 10. | Proper mounting locations will be created. |  |  |
| 3 | Mount all the components and make sure none interfere with each other. | All exterior components will be placed on enclosure. |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Name | Enclosure Meets PCB and Battery Test | | Test Number | NT\_03 | | | |
| Requirement(s) Tested | | 1.1 | Verification Method | I | A | D | T |
| Test Setup | | Finalized Circuit Board  Battery  Enclosure | | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Test Step | Action (Attach test data, diagrams, etc. as appropriate) | Expected Result | Observed Result | Pass  Fail |
| 1 | Place and secure battery in enclosure laying it flat on the bottom. | Battery should fit |  |  |
| 2 | Place and secure circuit board on top of battery. | Battery and circuit board should fit together |  |  |

|  |
| --- |
| Comments |

Date Test Engineer Witness

# Test Summary and Comments

The following table will be used as a reference for completed test plans.

Table Test Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Eng. Req. ID** | **Test Code** | **Test Name** | **Pass/ Fail** | **Comments** |
|  | ET\_01 | Microphone Test |  |  |
|  | ET\_02 | Amplifier Circuit Test |  |  |
|  | ET\_03 | Digital Switch Test |  |  |
|  | ET\_04 | Auxiliary Port and Cable Test |  |  |
|  | ET\_05 | Testing of Buttons and Switch |  |  |
|  | ET\_06 | Analog Input Electronics Test |  |  |
|  | ET\_07 | Audio Switch Test |  |  |
|  | ET\_08 | Analog Integration w ADC Test and Debug |  |  |
|  | ET\_09 | Power Consumption Test |  |  |
|  | ET\_10 | Audio Switching Function |  |  |
|  | ET\_11 | Interrupt Detection Device |  |  |
|  | ET\_12 | Acoustic Awareness Device |  |  |
|  | ET\_13 | Battery Power Consumption |  |  |
|  | ET\_14 | Acoustic Awareness Enabler 1.0 Debug Test |  |  |
|  | ET\_15 | Device Lifetime Test |  |  |
|  | ET\_16 | Device Improvement Methods |  |  |
|  | PT\_01 | Threshold Setting as f(σ) Concept Validation |  |  |
|  | PT\_02 | Standard Deviation Calculation |  |  |
|  | PT\_03 | Interrupt Calculation Process |  |  |
|  | PT\_04 | 20 Second Averaging, Interrupt Process |  |  |
|  | PT\_05 | Processing Debug Test |  |  |
|  | PT\_06 | Standard Deviation Debugging |  |  |
|  | PT\_07 | Device Accuracy Test & Debug |  |  |
|  | CT\_01 | ADC Test Code |  |  |
|  | CT\_02 | Audio Switch Test Code |  |  |
|  | CT\_03 | Setting Button Test Code |  |  |
|  | CT\_04 | Main µC Test Code |  |  |
|  | CT\_05 | Process Implement in µC |  |  |
|  | BT\_01 | PCB Test Layout |  |  |
|  | BT\_02 | Initial PCB |  |  |
|  | BT\_03 | PCB Review |  |  |
|  | BT\_04 | PCB Manufacture Summary |  |  |
|  | BT\_05 | PCB Function Test and Comparison |  |  |
|  | NT\_01 | Enclosure Selection |  |  |
|  | NT\_02 | Enclosure Outfit for Components, I/O |  |  |
|  | NT\_03 | Enclosure Integration with Battery and PCB |  |  |

# A screen shot of a computer Description automatically generatedUpdated Plan and Schedule

A close up of a white wall

Description automatically generatedA picture containing sky

Description automatically generated

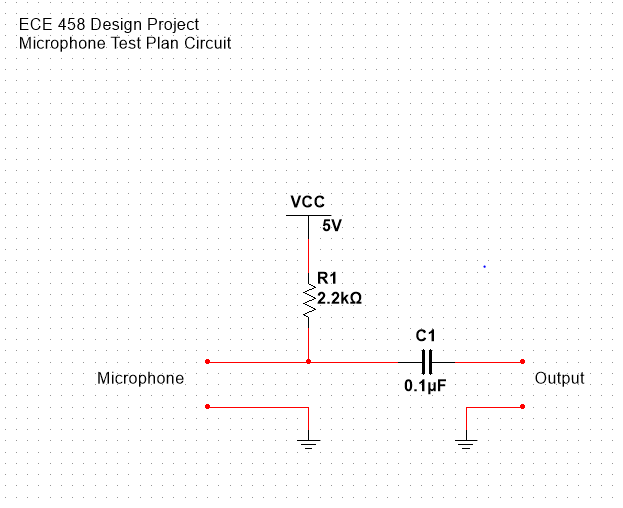


Figure 6 Microphone Circuit

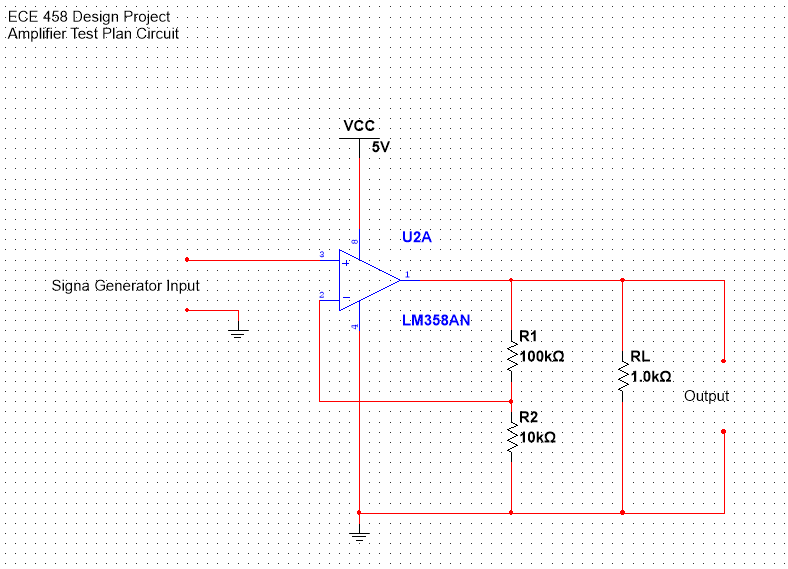


Figure 7 Amplifier Test Plan Circuit

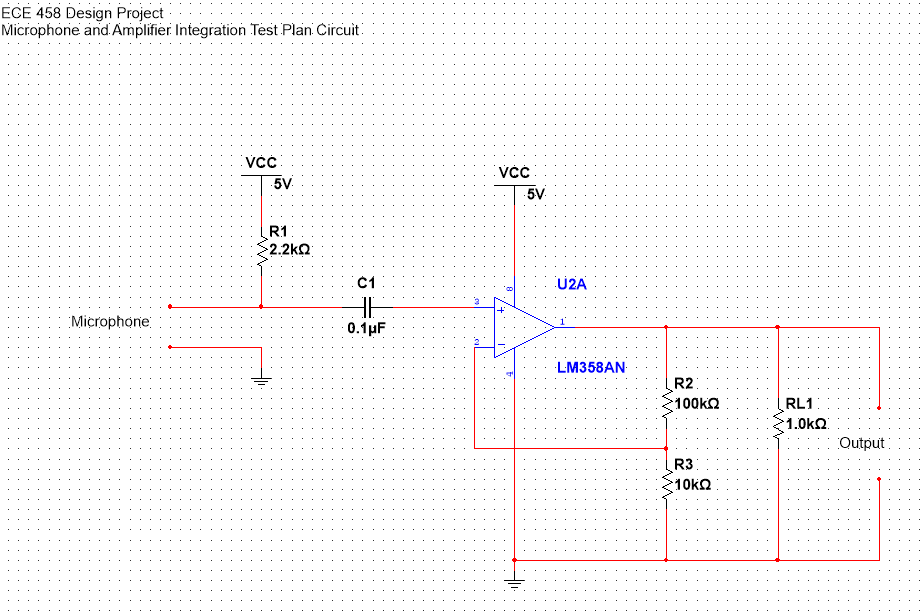


Figure 8 Microphone and Amplifier Integration Circuit

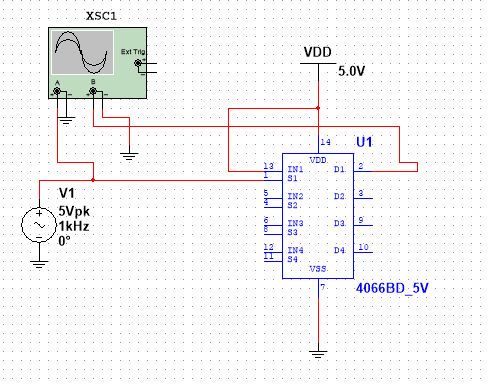


Figure 9 Digital Switch Testing Circuit

A close up of text on a white background

Description generated with high confidence

Figure Tentative Enclosure

## ADC Test Code (C)

|  |
| --- |
| /\* Senior Project: Test Program |
|  | \* Display ADC through LEDs |
|  | \* |
|  | \* The goal of this program is to connect the microphone to |
|  | \* the ADC on the microphone |
|  | \* left signal and right signal on and off simultaniously |
|  | \*/ |
|  |  |
|  | #include <avr/io.h> |
|  | #include <avr/interrupt.h> |
|  |  |
|  | volatile uint16\_t mic\_data \_\_attribute\_\_((address(0x800100))); |
|  |  |
|  | void clear\_ports() |
|  | { |
|  | PORTD &= 0x00; |
|  | PORTB &= 0x00; |
|  | } |
|  |  |
|  | int main() |
|  | { |
|  | mic\_data = 0; |
|  | DDRC &=~ (1 << DDC5); |
|  | // PORTC0 is the input from the Microphone |
|  | DDRB |= 0x03; |
|  | DDRD |= 0xFF; |
|  | clear\_ports(); |
|  | // PORTB1:0 and PORTD7:0 is the output of the Microphone. |
|  | ADMUX = (1 << REFS1)|(1 << MUX2)|(1 << MUX0); |
|  | ADCSRA |= (0b111 << ADPS0); |
|  | ADCSRA |= (1 << ADEN); |
|  | ADCSRA |= (1 << ADSC); |
|  |  |
|  | TCNT1 = 9999; |
|  | TCCR1A = (0b00<<COM1A0)|(0b00<<COM1B0)|(0b00<<WGM10); |
|  | TCCR1B = (0b00<<WGM12)|(0b101<<CS10); |
|  | TIMSK1 = (0b1<<TOIE1); // Enables interrupts for TOV1 |
|  |  |
|  | sei(); |
|  | while(1) |
|  | { |
|  | asm("NOP"); |
|  | } |
|  | } |
|  |  |
|  |  |
|  |  |
|  | ISR (TIMER1\_OVF\_vect) |
|  | { |
|  | clear\_ports(); |
|  | TCNT1 = 9999; |
|  | mic\_data = ADC; |
|  | PORTB = ((mic\_data >> 8) & 0x03); // Displays the highest 2 digits of the ADC |
|  | PORTD = mic\_data; // Displays the lowest 8 digits of the ADC |
|  | } |

Interrupt.cs

|  |
| --- |
| // Jared Alves |
|  | // Stephen Felix |
|  | // Michael Benker |
|  |  |
|  |  |
|  | using System; |
|  | using System.Linq; |
|  | using System.Collections.Generic; |
|  |  |
|  | namespace DSPSrDesign |
|  | { |
|  | class Program |
|  | { |
|  | static void Main(string[] args) |
|  | { |
|  | double two\_avg = 0; |
|  | double twenty\_avg = 0; |
|  | //double std\_20 = 0; |
|  | //double T1 = 0; |
|  | //double T2 = 0; |
|  | //double T3 = 0; |
|  |  |
|  | int time = 0; |
|  | int alltime = 0; |
|  |  |
|  | int A = 4; |
|  | double Threshold; |
|  | double sd = 0; |
|  |  |
|  | double[] inputdata\_two = new double[200]; |
|  | double[] inputdata\_twenty = new double[10]; |
|  |  |
|  | while (true) { |
|  |  |
|  |  |
|  |  |
|  | Console.WriteLine("Time is {0} seconds", alltime); |
|  |  |
|  | Console.WriteLine("Enter a minimum value (dB) between 30 and 90:"); |
|  | string Min = Console.ReadLine(); |
|  | Console.WriteLine("Enter maximum value (dB) between 30 and 90:"); |
|  | string Max = Console.ReadLine(); |
|  |  |
|  | int Min\_int = Int32.Parse(Min); |
|  | int Max\_int = Int32.Parse(Max); |
|  |  |
|  |  |
|  | Random randNum = new Random(); |
|  | for (int i = 0; i < inputdata\_two.Length; i++) |
|  | { |
|  | inputdata\_two[i] = randNum.Next(Min\_int, Max\_int); |
|  | } |
|  |  |
|  |  |
|  | two\_avg = inputdata\_two.Average(); //Get average of random data |
|  | Console.WriteLine("The two second average is {0}", two\_avg); |
|  |  |
|  |  |
|  | inputdata\_twenty[time/2] = two\_avg; |
|  |  |
|  | foreach (double item in inputdata\_twenty) |
|  | { |
|  | Console.WriteLine(item.ToString()); |
|  | } |
|  | Threshold= A\*sd + twenty\_avg; |
|  |  |
|  | if (alltime > 20) |
|  | { |
|  | if (two\_avg >= Threshold) |
|  | { |
|  | alltime = -2; |
|  |  |
|  | Console.WriteLine("Interrupt has occurred"); |
|  | } |
|  | } |
|  |  |
|  | if (alltime >= 20) |
|  | { |
|  |  |
|  |  |
|  | twenty\_avg = inputdata\_twenty.Average(); |
|  | Console.WriteLine("The twenty second average is {0}", twenty\_avg); |
|  |  |
|  | double sumOfSquaresOfDifferences = inputdata\_twenty.Select(val => (val - twenty\_avg) \* (val - twenty\_avg)).Sum(); |
|  | sd = Math.Sqrt(sumOfSquaresOfDifferences / inputdata\_twenty.Length); |
|  | Console.WriteLine("The twenty second standard deviation is {0}", sd); |
|  |  |
|  | } |
|  |  |
|  |  |
|  | Console.WriteLine("------------------------------------------\n \n"); |
|  | alltime = alltime + 2; |
|  | if (time < 18) |
|  | { |
|  | time = time + 2; |
|  | } |
|  | else |
|  | { |
|  | time = 0; |
|  | } |
|  |  |
|  |  |
|  | } |
|  | } |
|  | } |
|  | } |