**ECD423 Novel MIDI Instrument**

**ECD Integration and Test Plan**

# Scope

## System Overview

This document details the integration and test plan for the project. Successful system verification should demonstrate that the ECD 423 Novel MIDI Instrument is in full compliance with its Project Specification to all key stakeholders.

The purpose of this project is to create a Novel MIDI Instrument that is capable of polyphony, variable volume control and pitch bend. Our design is a detachable metal rail with 2 sliding keypads, a tilt detection system and a base station which is attachable to the lifting tube of the mic stand. The base station sends signals to the synthesizer via USB port. The device is capable of being attached to a tripod boom mic stand without inhibiting the mic stand's original functionality. The user is able to play a variety of MIDI instruments on the device. Keypad rotation, keypad location on the rail and tilting of the rail is taken into account in order to switch between different instruments. The user is able to control the type of notes being produced by the synthesizer using keys on the keypad as well as location of the keypad on the rail. Sliding the keypads up and down the rod can produce note changes depending on what instrument is being played. A joystick attachment to the side of each keypad is controlled by the user’s thumb and can be used to produce pitch bend.

## Document Overview

This test plan describes the test site(s) and test environment(s) to be used in Section 3 and the tests in Section 4, followed by the detailed test procedures in Section 5.

In this document, a mandatory practice is identified by “shall”, a good practice by “should”, permission by “may” or “can”, expected outcome or action by “will”, and descriptive material by “is” or “are” (or another verb form of “to be”).

# Referenced Documents

The following documents are referenced and inform part of this document. The table of Finalized Derived Requirements which was finalized in the Project Specification document, included in Appendix A of the Final Report is also included in this document in Appendix A.

* + - 1. ECD 423 Novel MIDI Instrument Final Report
      2. ECD 423 Novel MIDI Instrument Project Specification

# Integration and Test Environment

## Seymour Kunis Media Core

The primary site for the tests outlined in this document is the Seymour Kunis Media Core room, situated in the Engineering and Science building at the Innovative Technologies Complex. The selection of this room was influenced in part by the endorsement of the team's advisor, Dr. Scott Craver, and also because it is equipped with all the essential tools required for constructing and testing the project, including synthesizers, 3D printers, and previous MIDI projects.

### Hardware and Firmware Items

The list below reveals the hardware components and applications used during the testing stage to demonstrate completion of design requirements.

**Testing Equipment**

1. 3D Printer: The majority of components in this project will be 3D printed due to its flexibility in creating different shapes and sizes for the keypad. This device will be used continuously during the design process to determine and optimize the dimensions of the keypad components..
2. Multimeter: This device is used to measure the current/voltage of specific components that are connected on the breadboard. It is used to locate errors within circuitry and to aid in creating solutions for these problems.
3. Oscilloscope: This is used for similar reasons as the multimeter as it generates visual representations of voltage signals over time which can display the strength of the wireless communicator over specific distances.
4. Power Supply: The keypads will need a constant flow of 3.3V to power each keypad in order to maintain regular functions. A strong power source is crucial to implementing any tests that involve operating the keypads.
5. Weights: In order to test and calibrate the force sensitive resistors, small weights will be put on the key to apply the constant force which should be interpreted relatively the same on each force sensitive resistor.

**Project Equipment**

1. Keypad Enclosure: This device will be tested to show a number of requirements for this project. The first being how accurate the data, which is user pressure, is reflected based on the change in volume in the Synthesizer. The second being the ability of the system to determine in what rotated state the keypads are in at any given time. The third is to check whether the lamello-like keys are able to operate like piano keys. Finally, the response time test will utilize this hardware component to determine time taken from the user pressing on a key to a MIDI sound being produced.
2. Tilt Detection System: This system is a separate part of the project intended to detect 3 different states that the mic stand will be in at any point in time. There will be a test specifically conducted to prove this function by attaching LEDs in series with each tilt switch and using a protractor to determine the range of angles which correspond to each state (horizontal, vertical, and angled).
3. Rotary Encoder: This piece will be built-in to the lower segment of the keypad and will be used to determine the distance of each keypad from the center of the mic stand at any given point in time. Through the rotary encoder Test, the device will be tested for accuracy in calculating relative position when the user moves the keypad along the rail.
4. nRF24L01: This is a wireless transceiver/receiver module whose purpose in this project is to provide wireless communication between the keypad and the base station (Arduino Nano/Teensy to the Teensy 4.0). It relays information unidirectionally by acting as a bridge to relay user input from the keypad to trigger corresponding sounds from the synthesizer.
5. Laptop: Laptops will be used to program Arduinos and read the serial monitor off arduino. The laptop will also be used to host the element program to which the arduino will connect to via usb.
6. Force Sensitive Resistors: In order to measure the press of a key, a force sensitive resistor will be used. when connected to an arduino it will give an 8 bit value of pressure exerted.
7. Joystick: This is an add-on piece to the side of the keypads and will be used to control pitch bend. moving the stick up or down will change the pitch of the note. This will be used in the Joystick Test which will look for the ability to control pitch bend.

### Software Items

The list below reveals the software components and applications used during the testing stage to demonstrate completion of design requirements.

1. Autodesk Fusion 360: This application is used to produce CAD designs such as the keys on the keypad, housing design, and sliders which attach the keypad to the metal rod.
2. Element (Synthesizer): This software application is the final destination for the MIDI signal. It is what produces the particular note that the user would like to hear. User data is produced and processed through Arduino Nano’s/Teensy’s to the Teensy 4.0 wirelessly which then gets converted into usable MIDI signals to the synthesizer.
3. Arduino IDE: This is a code editor to connect with Arduino hardware and give it specific instructions. The entire software application of this project relies heavily on this interface by programming Arduino microcontrollers with code that allows them to convert information to usable data.
4. RF24 Library: This library was installed onto the Arduino IDE for the purposes of this project. It has built in functions which are used to establish communication between the transceiver and receiver(nRF24L01) and transmit signals in one direction, from the Arduino Nano/Teensy to the Teensy 4.0.
5. Control Surface Library: Similar to the RF24, this library was also installed for this project for its pre-existing functions. These functions are implemented by the Teensy 4.0 and sent to the synthesizer such that it can trigger specific notes, generate pitch bend and perform variable volume control.
6. File demo1\_MidiTest: This file contains the Arduino code to send MIDI signals from the base station microcontroller to the synthesizer. It contains a set of for loops to demonstrate the playing of notes, pitch bend, variable volume control and changing of instruments.
7. Files demo2\_distanceTestTx and demo2\_distanceTestRx: This file contains code written in Arduino IDE to continuously send a fixed message from the transceiver to the receiver. It will be used in the wireless distance test to determine whether the distance between wireless communication is met.
8. File demo3\_multipleTransceiver: This file contains code written in Arduino IDE to continuously send messages between 2+ transceivers uni-directionally towards one receiver. It is a test of the quality of the message as well as the speed at which the message is being transmitted.
9. File demo4\_KeypadOrientation: This code contains serial print messages to tell the tester what orientation the mic stand is based on the tilt detection system. It is to be used when the microcontroller is being connected to the circuitry with the tilt sensors and will display the horizontal, vertical, or angled orientation. In addition, the relative position of the rotary encoder and the rotation of the keypad is displayed here as well. The instrument that the mic stand is switched to depending on the orientation of these 3 factors is also printed out.
10. Input Files: A collection of .ino files which are used to program onto the microcontrollers in this project through Arduino IDE. These files are used to set up connections between hardware components to accurately process data.
11. Keypad Program: This program that will run on the keypad microcontrollers This program will get all the data from the various sensors on the keypad and pack them into easily transmittable messages.
12. Base Station Program: This is the program that will run on the base station microcontroller. This program will receive the messages from the keypads and then use that to create a midi message and send it to the laptop.

### Other Materials

No Other Materials except the materials stated above will be needed for the tests.

### Participants

Tests will be conducted at the Media Core lab by the members of the ECD 423 Project.

### Tests To Be Performed

1. MIDI Test: This test will focus on sending MIDI messages from the Base Station to a synthesizer called Element. It will be conducted with a Teensy 4.0 connected over USB to a synthesizer program. The Teensy will be programmed to send specific MIDI  
   messages to the computer and the MIDI monitor on the synthesizer program will be checked to determine whether the correct MIDI notes are being emitted along with the correct pitch bend and volume.
2. Wireless Communication Distance Test: This evaluation will be performed to determine whether the connection distance between transceiver and receiver modules is reliable enough to prevent disconnect. Also, there should be a transmittable distance of at least 50 inches.
3. Multiple Transceivers Communication Test: The purpose of this evaluation is to demonstrate the ability to perform uni-directional communication by 2 transceivers and 1 receiver module. The quality/frequency of successful transmissions will be taken into account as well.
4. Tilt Sensor Test: This test will test to see if the tilt sensors can accurately measure the 3 positions of tilt (vertical, horizontal and slanted) while the device is in use.
5. Rotary Encoder Test: This test will be used to determine how accurate the rotary encoder can measure the relative position of the keypad on the rod. It will be conducted by moving a rotary encoder with a wheel attachment along the rail. The rotary encoder will be connected to an Arduino Nano/Teensy to record measurements obtained from the encoder.
6. Keypad Rotation Test: This test will ensure that both keypads are able to detect the 4 positions of rotation, which include .
7. Instrument Transition Test: This test will determine if the instrument can accurately detect all of the different instruments that the user is attempting to play.
8. Variable Volume Test: This test focuses on the amount of force the Force Sensitive Resistors (FSR) can read from the keypresses. This test will be performed by applying forces to the key with an FSR underneath and it can reliably detect 10 different levels of force which will correspond to volume percentage. The keys will be hooked up to an analog multiplexer which will be controlled and read by an Arduino Nano/Teensy.
9. Lamellophone Key Test: This test will determine if the lamellophone key can operate a force sensitive resistor. It will consist of the lamellophone portion being plucked and whether the FSR can reliably detect 10 different levels of force in this mode.
10. Joystick Test: This test will be performed to check that the joystick can be read and translated into a pitch bend message. The test will consist of a two-axis joystick being connected to an Arduino Nano/Teensy and that the Nano/Teensy can reliably read the output of the two-axis joystick.
11. Final Test: This is an evaluation on the product as a whole. It will involve playing a tune on the MIDI mic stand to demonstrate its usability. The user will also demonstrate switching between different instruments.

# System Integration

## System Integration Diagram

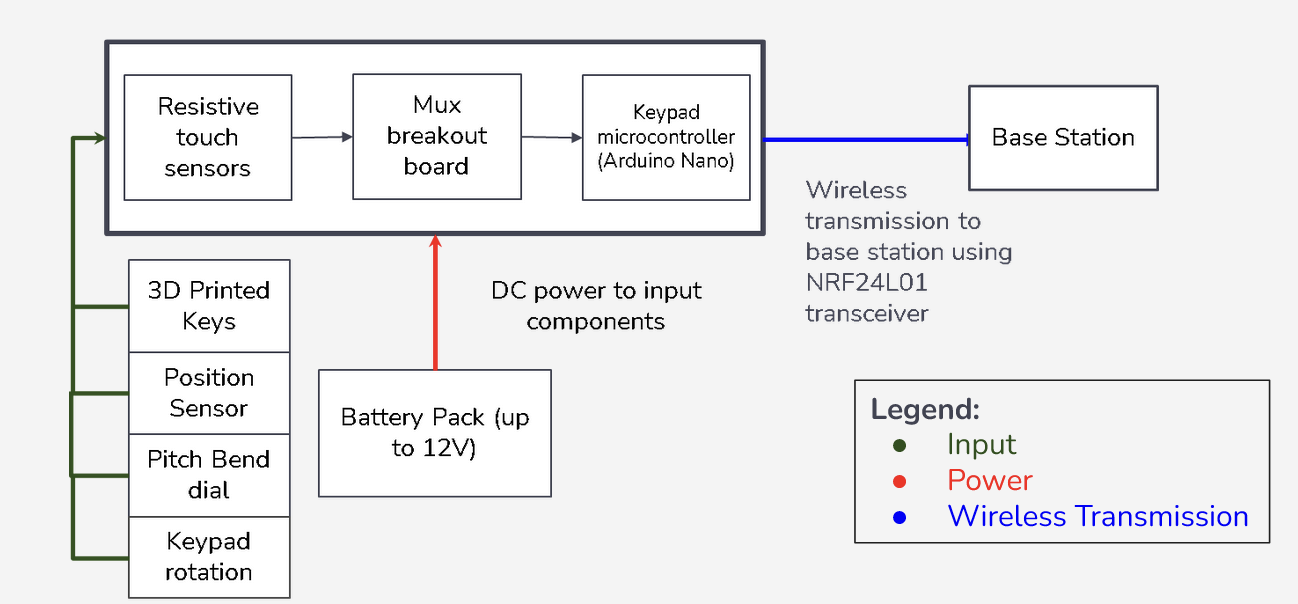


Figure 1. Diagram detailing the keypad input

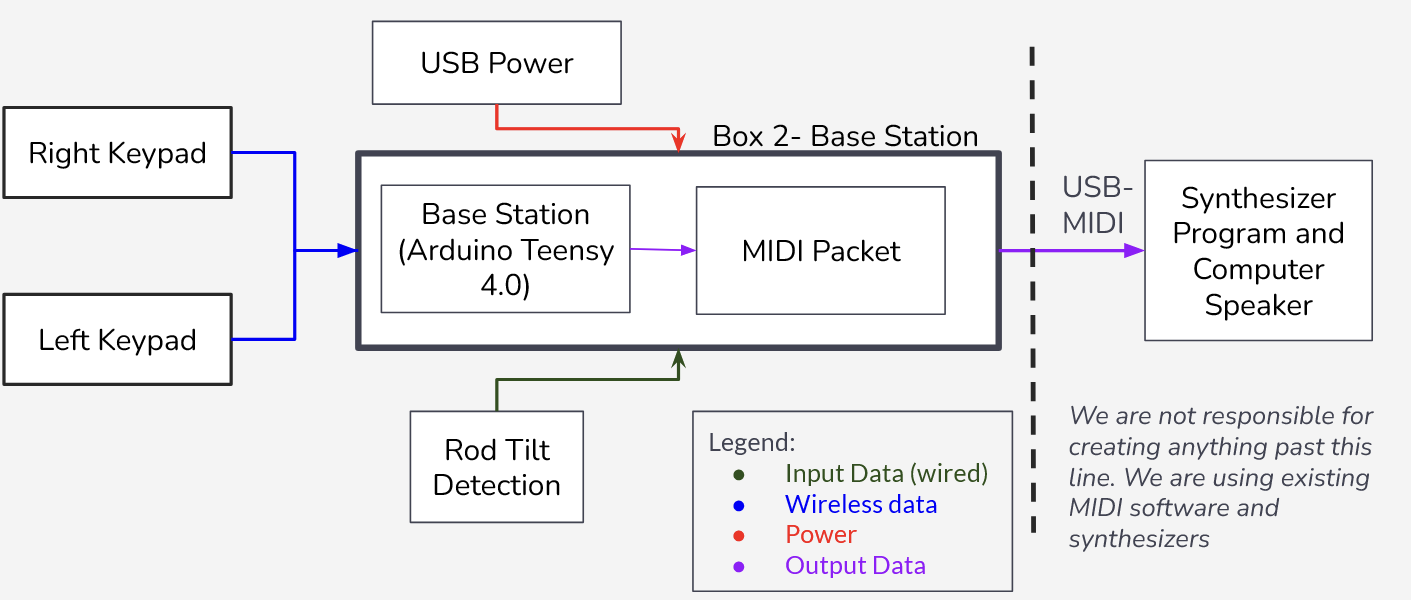


Figure 2: Diagram detailing the operation of the base station

## User Input and Circuit

The user input is taken from a set of 7 keys on each keypad. Beneath each of the keys is a force-sensitive resistor (FSR). By pressing the key harder or softer, the user is able to control the volume of the MIDI output signal. All the FSRs in each keypad are connected to a multiplexer (MUX) breakout board, which allows multiple signals to be connected to the input pins on the Arduino board. In this way, the number of input pins can be greatly reduced and encoded into a smaller number of output pins.

## Circuits and Software

The multiplexer circuit and the Arduino Nano/Teensy in the keypads will be configured in such a way where one pin on the Arduino accepts the input and outputs it from another pin to the nRF24L01 wireless transmitter. The wireless transmitter sends this data to the Arduino Teensy 4.0 board inside the base station, as detailed above.

## MIDI Software

Once the multiplexed signals are received by the base station, they must be converted, along with the pitch bend data from the joystick on the side of the keypad to a MIDI signal. Once synthesized, the MIDI packet is sent to the open-source synthesizer software Element, where it can be processed and outputted.

* 1. Output

After being processed in the Element software, the synthesized MIDI packet can be output over a set of computer speakers, at which point the project requirements are satisfied.

# System Verification

These qualification methods will be used in verifying that the project system meets all requirements in its Project Specification.

* Demonstration (D)
* Test (T)
* Analysis (A)
* Inspection (I)

The following Requirement Categories (RC) are used to describe the purpose of each test:

* System Capability Requirements (SC)
* System External Interface Requirements (EI)
* Project Business Requirements (PB)
* Other Requirements (O)

## Verification Overview

The table displayed below shows all of the tests to be performed as proof of meeting all design requirements mentioned in the project specifications document. The name of the test is provided along with their Test ID, details regarding the requirement being fulfilled and test completion date. Please note that test completion dates which display times that have yet to arrive are simply estimated dates and not the actual time of completion.

## Test Coverage

| **Test ID** | **Test Name** | **QM** | **RC** | **Requirements Addressed** | **Test Completion Date** |
| --- | --- | --- | --- | --- | --- |
| ECD423-T-001 | MIDI Test | I,D,T | O,EI,SC | ECD423-R-001  ECD423-R-002  ECD423-R-003  ECD423-R-004  ECD423-R-011 | 2023-11-13 |
| ECD423-T-002 | Wireless Communication Distance Test | I | O,SC | ECD423-R-004  ECD423-R-008 | 2024-02-09 |
| ECD423-T-003 | Multiple Transceivers Communication Test | I,T | O, SC | ECD423-R-008  ECD423-G-001 | 2024-03-01 |
| ECD423-T-004 | Tilt Sensor Test | A,I | SC | ECD423-R-009  ECD423-R-010 | 2024-03-08 |
| ECD423-T-005 | Rotary Encoder Test | D | SC | ECD423-R-006  ECD423-R-007  ECD423-R-009  ECD423-G-004 | 2024-03-08 |
| ECD423-T-006 | Keypad Rotation Test | I,D,A | SC | ECD423-R-005  ECD423-R-006  ECD423-R-007  ECD423-R-009  ECD423-G-004 | 2024-03-15 |
| ECD423-T-007 | Instrument Transition Test | D | SC | ECD423-R-007  ECD423-G-004 | 2024-03-22 |
| ECD423-T-008 | Variable Volume Test | D,I | SC, O | ECD423-R-003  ECD423-R-004 | 2024-03-01 |
| ECD423-T-009 | Lamellophone Key Test | D,I | SC | ECD423-R-003  ECD423-G-005 | 2024-03-08 |
| ECD423-T-010 | Joystick Test | D | SC | ECD423-R-002 | 2024-03-15 |
| ECD423-T-011 | Final Test | D | SC, EI | All Requirements | 2024-03-22 |

**6 System Verification Tests**

## 

### Communications Test Group Overview

This test group includes all wireless communication tests and the outputting of MIDI messages.

### Test Procedures

#### ECD423-T-001 MIDI Test

This test is implemented with the help of the Teensy 4.0 device which is connected to a computer via USB connection. The purpose of this evaluation is to demonstrate the ability to send MIDI signals which emit notes with varying variable volume control and pitch bend. Additionally, the program should switch between different channels. This is done by programming the microcontroller with code from Arduino IDE. This code utilizes built in functions from the Control Surface Library to create for loops which demonstrate scaling of the notes ranging from 0-127. An additional two for loops are used to show variable volume ranging from values 0-127 and pitch bend. The results of this program are shown through the monitor on the synthesizer program, Element.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect Teensy 4.0 to the computer via USB connection. | No results expected as this is setup |  |
| 2 | Open, verify and upload the program used to demonstrate this test. | The Arduino IDE Serial Monitor should display which channels are currently on/off at any given time in addition to their volume, and note being sent. |  |
| 3 | Open the synthesizer program, Element. Go to File>Preferences>MIDI and turn “Active MIDI inputs” on. | All current results are maintained with no added changes made. |  |
| 4 | Turn your volume to the appropriate range and start observing the MIDI monitor. | There should be a total of 3 MIDI monitors displaying the current note, pitch bend, and volume control at any given time on the synthesizer. |  |

#### ECD423-T-002 Wireless Communication Distance Test

This test will ensure that both keypads can send wireless messages to the base station reliably without transmission errors. Additionally, the wireless modules must be able to maintain communication of up to 50 inches. This test consists of two Arduino Nano’s/Teensy with nRF24L01 transceivers transmitting data and 1 Teensy 4.0 with a nRF24L01 transceiver receiving the data. The transmitters will slowly be moved away from the source until the receiver can no longer reliably receive data. An evaluator will stop and record the distance between modules once the connection has been broken.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the 2 Nano’s/Teensy’s and 1 Teensy 4.0 to their respective nRF24L01 modules. | No results expected as this is setup |  |
| 2 | Connect the Teensy 4.0 to one laptop and both Nano’s/Teensy’s to another. Note, for this experiment the Nano’s/Teensy’s and Teensy must not be using the same laptop unless you have cables that can extend at least 50 inches in width. | No results expected as this is setup |  |
| 3 | Open, verify and upload the code for the transceivers to the Nano’s/Teensy’s and the code for the receiver to the Teensy. | Serial Monitor should display that the program has finished set up. |  |
| 4 | Have person A look at the Serial Monitor on the Arduino IDE and person B to move the Nano’s/Teensy’s away from the Teensy at a steady pace. When the connection is broken, person A will tell person B to stop. Person B will measure the distance from the Teensy to the Nano/Teensy. | Person A should see on the Serial Monitor for the receiver code, the message being received. When the message to be received is displaying “No msg received” for an extended period of time. Person B should record the distance between the wireless modules and check to see that the distance is at least 50 inches apart. |  |

#### ECD423-T-003 Multiple Transceivers Communication Test

This is a test performed with the sole purpose of connecting two transceiver modules to a single receiver module for constant uni-directional communication. Each transceiver module will be attached to each keypad respectively and the receiver module will be located at the base station. The strength of transmission signals will be recorded based on the frequency of failed transmission. The average time to transmit each signal will be taken into consideration in regards to evaluating the quality of this communication.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the 2 Nano’s and 1 Teensy 4.0 to their respective nRF24L01 modules. | No results expected as this is setup |  |
| 2 | Connect the Teensy 4.0 to one laptop and both Nano’s to another. Note, for this experiment the Nano’s and Teensy must not be using the same laptop unless you have cables that can extend at least 50 inches in width. | No results expected as this is setup |  |
| 3 | Open, verify and upload the code for the transceivers to the Nano’s and the code for the receiver to the Teensy.  Make sure to type “R” into the Serial Monitor for the receiver and different node addresses, from 0-5, for each transceiver. | The Serial Monitor for the receiver should recognize it is in receiving mode and that of the transceiver should start sending transmission signal’s with payload ID. |  |
| 4 | Observe the Serial Monitor for both transceivers and determine the average time to transmit signals for each. Take note of the max value observed from the Serial Monitors. | The transceiver should be taking on average 580-600us to transmit signals. single transmission time should not exceed 10ms. |  |
| 5 | Have one person continuously read the payload ID on the receiver side to determine if there are any missed transmission signals. | The receiver should be receiving almost all of the messages being sent. This can be observed since the payload ID’s for each transmission signal increases by a factor of 1. |  |

## 

### Position Test Group Overview

This test group handles the tests that require physical adjustments to the instrument.

#### ECD423-T-004 Tilt Sensor Test

The test will be performed by attaching our tilt sensor to the microphone stand and then tilting the rod of the stand to desired tilt level and seeing if the correct value is read. Light shaking will be applied to the rod also at each tilt position to make sure no false readings occur. The tilt sensor will be attached to a Teensy 4.0 which will read the sensor. The results of this test will determine how accurate our tilt sensor can be to detect current position.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the tilt sensors to the Teensy 4.0 base station’s digital inputs, 3.3V output and ground. | No results expected as this is setup. |  |
| 2 | Tilt the microphone stands in a horizontal position. | Teensy outputs that the microphone stand is horizontal to serial monitor. |  |
| 3 | Lightly shake the microphone stand in the horizontal position. | Teensy outputs that the microphone stand is horizontal to serial monitor. |  |
| 4 | Tilt the microphone stand to the tilted position. | Teensy outputs that the microphone stand is tilted to the serial monitor. |  |
| 5 | Lightly shake the microphone stand in the tilted position | Teensy outputs that the microphone stand is tilted to the serial monitor. |  |
| 6 | Tilt the microphone stand to the vertical position. | Teensy outputs that the microphone stand is vertical to the serial monitor. |  |
| 7 | Lightly shake the microphone stand in the vertical position. | Teensy outputs that the microphone stand is vertical to the serial monitor. |  |

**ECD423-T-005 Rotary Encoder Test**

This test will test the accuracy of the rotary encoder. The rotary encoder will start from a known position and then have a counter on the microcontroller count if the rotary encoder turns left or right.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the rotary encoder to 2 digital pins on the keypad microcontroller. | No results expected as this is setup. |  |
| 2 | Start the rotary encoder at one end of the rod and set the internal count inside the microcontroller to 0. | No results expected as this is setup. |  |
| 3 | Move the keypad to the other end of the rod. | The value on the internal counter should increase. |  |
| 4 | Move the keypad back to its initial position | The value on the internal counter should decrease. |  |
| 5 | Evaluate the counter at the initial position. | The value on the internal counter should read 0. |  |

#### ECD423-T-006 Keypad Rotation Test

This test will have contacts placed in each of the four positions on the baseplate and a reader of the contacts will be on the top plate. The reader will be connected to an Arduino Nano/Teensy to read the results. The results of this test will ensure that the rotation system is working correctly.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect rotation clutch contacts to 4 digital inputs on the Arduino Nano/Teensy. | No results expected as this is setup. |  |
| 2 | Rotate the clutch so that it is in the 1st position. | Arduino Nano/Teensy outputs that it is in the 1st position on the Serial Monitor. |  |
| 3 | Rotate the clutch so that it is in the 2nd position. | Arduino Nano/Teensy outputs that it is in the 2nd position on the Serial Monitor. |  |
| 4 | Rotate the clutch so that it is in the 3rd position. | Arduino Nano/Teensy outputs that it is in the 3rd position on the Serial Monitor. |  |
| 5 | Rotate the clutch so that it is in the 4th position. | Arduino Nano/Teensy outputs that it is in the 4th position on the Serial Monitor. |  |

#### ECD423-T-007 Instrument Transition Test

The test will utilize the established tilt sensors, rotary encoder, and keypad rotation mechanisms. The software will use the detected rod tilt, rod position, and keypad rotation to determine the correct instrument setting. The results of this test will ensure that the MIDI instrument is transitioning to the correct instrument.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect tilt sensor to Teensy 4.0 base station’s digital inputs, 3.3-volt output and ground and connect Rotation clutch contacts to 4 digital inputs on the Arduino Nano/Teensy. Maintain the keypads of the instrument in its horizontal position. | The default position of the instrument is the piano setting, and the instrument “Piano” will output on the serial monitor and the Element software will use the correct MIDI packages. |  |
| 2 | Invert the left-hand keypad and maintain the horizontal position of the rod. | The instrument name “Flute” will output on the serial monitor and the Element software will use the Flute MIDI packages. |  |
| 3 | Turn the rod vertically and lock the bottom keypad in its upright position. The top keypad will be free-sliding and upside-down. | The instrument name “Bass” will output on the serial monitor and the Element software will use the Bass MIDI packages. |  |
| 4 | The rod will maintain the vertical position. Rotate the upper keypad facing left and the bottom keypad facing right. Lock the keypads in place. | The instrument name “Bassoon” will output on the serial monitor and the Element software will use the Bassoon MIDI packages. |  |
| 5 | Position the rod of the instrument at a 45 degree angle. The keypads will be close together in the middle of the rod. Rotate the left keypad to its default rotation, and invert the right keypad. | The instrument name “Clarinet” will be output on the serial monitor and the Element software will use the Clarinet MIDI packages. |  |
| 6 | Move the keypads further apart from the clarinet test’s position | The instrument name “Saxophone” will output on the serial monitor and the Element software will use the Saxophone MIDI packages. |  |

## 

### Inputs Test Group Overview

This test group handles the tests for all user inputs.

#### ECD423-T-008 Variable Volume Test

The test will have force sensitive resistors positioned under the keys. The force sensitive resistors are connected to the analog inputs of the Arduino Nano/Teensy, utilizing the multiplexer breakout boards for additional inputs.The force sensitive resistors will output a numerical value to the serial monitor. This numerical value will be divided into ten ranges and used by the Element software to control the variable volume. The results of this test will ensure that the MIDI instrument is capable of changing volume based on the force applied to the input keys.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the keypads and force sensitive resistors to the Arduino Nano’s/Teensy’s 3.3-volt output, ground, and analog inputs. | The serial monitor will output 0 for each of the force sensitive resistors, as no force is being applied. |  |
| 2 | Apply varying levels of force to each of the keys. | The serial monitor will output a value in the ten ranges from 0 to 1023 for the resistor being pressed. The instrument will play in the corresponding volume level. |  |

#### ECD423-T-009 Lamellophone Key Test

The test will use the lamellophone keys that extrude out of the piano keys. The lamellophone keys allow for a strumming and plucking motion for instruments that do not use a simple press. The lamellophone uses the same force sensitive resistors as the connected piano keys. The results of this test will ensure that the lamellophone keys can operate a force sensitive resistor.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the keypad and force sensitive resistors to the Arduino Nano’s/Teensy’s 3.3-volt output, ground, and analog inputs. | The serial monitor will output 0 for each of the force sensitive resistors, as the lamellophone keys are in a neutral position. |  |
| 2 | Configure into an instrument position that utilizes the lamellophone keys. | The serial monitor will output 0 for each of the force sensitive resistors, as the lamellophone keys are in a neutral position. |  |
| 3 | Pluck the lamellophone keys with varying levels of force. | The serial monitor will output a value in the ten ranges from 0 to 1023 for the resistor being pressed. The instrument will play in the corresponding volume level. |  |

#### ECD423-T-010 Joystick Test

The test will use the two-axis joystick in its vertical axis. The joysticks are attached on the side of the keypads, positioned under the user's thumbs. The joysticks will be connected to the Arduino Nano/Teensy and the value of the position will be used by the Element software to control pitch bend. The results of this test will ensure that the MIDI instrument is capable of pitch bend.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the joysticks to the Arduino Nano’s/Teensy’s 3.3-volt output, ground, and analog inputs. | The serial monitor will output approximately 512 at the resting neutral position. |  |
| 2 | Steadily move the joystick upwards. | The serial monitor will gradually output a higher number, reaching the maximum value of 1023 at the highest point. The pitch will bend up. |  |
| 3 | Return to the neutral position. Steadily move the joystick downwards. | The serial monitor output will gradually decrease, reaching the minimum value of 0 at the lowest point. The pitch will bend down. |  |

#### ECD423-T-011 Final Test

This evaluation is to demonstrate the functionality of the final product as a MIDI instrument capable of pitch bend, variable volume and polyphony. The user will perform a simple tune/song and alternate between at least 6 different instruments.

| **#** | **Step-by-Step Operations** | **Expected Results** | Actual Results |
| --- | --- | --- | --- |
| 1 | Connect the base station to a laptop with the synthesizer program, Element, already on. | No results expected as this is setup. |  |
| 2 | Attach the keypads and tilt detection system onto the rod and then attach the rod onto the mic stand. Adjust the height of the mic stand appropriately. | No results expected as this is setup. |  |
| 3 | Change between MIDI instruments by changing the orientation of the rod, positioning of the keypad and keypad rotation. | Pressing on the keys should emit a sound that is reminiscent of the particular instrument the device is set to. |  |
| 4 | Play a tune. | The device should sound similar to the actual instrument. |  |

**Appendices**

# Appendix A

**Final Derived Requirements**

| ID | QM | RC | Derived Requirement |
| --- | --- | --- | --- |
| ECD423-R-001 | D | SC | The MIDI instrument shall be able to produce polyphony. |
| ECD423-R-002 | D | SC | The MIDI instrument shall be able to produce pitch bend. |
| ECD423-R-003 | D | SC | The MIDI instrument shall be able to produce variable volume control. |
| ECD423-R-004 | I | O | It shall output accurate MIDI signals to a synthesizer. |
| ECD423-R-005 | I | SC | The device shall not inhibit the functions of an ordinary microphone stand. |
| ECD423-R-006 | D | SC | Both keypads will be capable of rotation and sliding movement on the rod. |
| ECD423-R-007 | D | SC | The device shall be able to switch between at least 6 different instruments based on keypad rotation, keypad location on rod, and rod tilting motion. |
| ECD423-R-008 | I | SC | There shall be wireless communication between the keypads and base station. |
| ECD423-R-009 | A | SC | The device shall be able to detect the position of both keypads on the rod at any given time. |
| ECD423-R-010 | I | SC | The base station shall have the capability to differentiate between 3 positions on the rod: horizontal, vertical, and angled. |
| ECD423-R-011 | T | EI | MIDI signals shall travel from the base station to a synthesizer via USB connection. |

**Stretch Goals**

| ID | QM | RC | Stretch Goals |
| --- | --- | --- | --- |
| ECD423-G-001 | T | O | The response time between note pressing and note emission from the synthesizer should be less than 10ms. |
| ECD423-G-002 | I | EI | The instrument should be able to synthesize sound directly from the product. |
| ECD423-G-003 | I | SC | Both keypads should have an electronic locking/unlocking mechanism to secure its position on the rod when locked and allow for movement when unlocked. |
| ECD423-G-004 | D | SC | The device should be able to switch between at least 10 different instruments based on keypad rotation, keypad location on rod, and rod tilting motion. |
| ECD423-G-005 | I | SC | The device should have 4 lamellophone style keys which are attached to force sensors and produce sound reminiscent of strumming string instruments. |