

Work Package

Chase Williams

- What is the work that needs to be performed?
 - Work description:
 - Skill set and level of experience
 - We need someone with intermediate experience with Arduino hardware and software. This means that they have worked on a few projects with these items previously.
 - However, we can also have one individual that knows the hardware side well and another individual that knows the software side well.
 - Intermediate C++ experience
 - General knowledge required
 - Needs to understand the basics of capacitors
 - Needs to know how to program
 - Needs to know how to set up a circuit on a breadboard.
 - Specific knowledge needed
 - Needs to know how to program in the Arduino IDE using C++
 - Specific deliverable or task
 - We need to figure out how to detect a touch on a nonmetallic keyboard key using capacitive sensors that are under the key. Essentially, we are wanting to create a proximity sensor to detect a touch, which does not require one's finger to complete the circuit to a capacitive touch key on the sensor. This task would help determine the capacitance threshold that indicates a touch occurred.
- What resources are needed to perform the work?
 - Materials
 - Capacitive 12 pin sensor – MPR121
 - 22 AWG copper wires with insulation
 - Aluminum tape
 - Arduino Nano
 - Arduino IDE
 - Breadboard
 - Solder
 - Soldering Iron
 - Soldering station
 - 2 - PCB mount headers
 - Mini-B to Type-A USB cable

- Computer with a USB port
 - Keyboard with removeable keyboard keys
- Training
 - Depending on the experience of the individual working on this project, we might need training on how to integrate the MPR121 sensor and the Arduino Nano.
 - Depending on the experience of the individual working on this project, we might need training on what Arduino code to use in order to get capacitance value readings.
- Provided Requirements
 - The distance between the wire connected to the key on the capacitive sensor and the top of a key.
 - The allowed range of error (in mm) of a touch on the keyboard key. This measures how far someone's finger is above the key, when a touch is detected by the program.
 - The time it takes for the software to show a touch was detected after the physical touch was made needs to be less than 70ms.
- Provided Specifications
 - Operating voltage and current of the MPR121 capacitive sensor
 - Operating voltage and current of the Arduino Nano
- Testing Plan
 - We would show our plan of an iterative approach on testing when a finger touched a key. This would involve one touching a key and measuring the capacitance to determine the threshold that indicates a touch occurred on the key.
- What are the steps (flow) of the work?
 - When opening the work page
 - The individual(s) working on this project will need to read through the provided requirements, specifications, and the testing plan.
 - Once the individual(s) understand the above, they will need to go through the trainings if necessary.
 - While performing the work
 - Set up the code in the Arduino IDE that will be uploaded to the Nano
 - The code prints the capacitance readings
 - Solder the PCB mount headers onto the capacitive sensor.
 - Attach the capacitive sensor to the breadboard.
 - Attach the Arduino Nano (which already has the PCB mount headers connected to it) to the breadboard
 - Make the necessary connections between the Arduino Nano and the capacitive sensor using the wires.
 - Vin on MPR121 to 5V on Arduino Nano
 - GND on MPR121 to GND on Arduino Nano

- SCL on MPR121 to A4 on Arduino Nano
 - SDA on MPR121 to A5 on Arduino Nano
- Connect a wire to a capacitive sensing node on the capacitive sensor
- One the other end of the wire, attach tape to it
- Remove any one key from the physical keyboard
- Attach the wire with tape to the bottom of a keyboard key
- Connect the Nano to the computer using the USB cable
- Upload the Arduino IDE code onto the Arduino Nano
- Open the serial terminal and make sure the capacitive sensor is recognized
- Read the capacitance values coming from the terminal when your finger is touching the key and note the values
- After completion of work
 - Find the typical threshold that determines a touch was made on the key
 - Add this threshold the Arduino code to indicate when a key was touched
 - Give 12 keyboard keys the capacitive sensing technology
 - Test how accurate the threshold is
- How is performance measured for the effort?
 - Performance in general will be measured by how long it takes to have functioning hardware, code, and when the capacitance threshold for a touch is determined.
 - The hardware needs to be set up and functioning for 1 key in 1 day. This, of course, does not factor in the software that will be uploaded to the Arduino Nano.
 - The hardware needs to be set up and functioning for all 12 keys in 1 day.
 - The software should output the capacitance readings for one capacitive key on the sensor in 3 days.
 - The software should output the capacitance readings for all 12 capacitive keys on the sensor in 2 days.
 - It should take 4 days to determine a capacitance threshold that accuracy (95% of the time) predicts when someone's finger is touching a key. This accounts for multitouch scenarios when multiple keys are being touched.
- How is the work-product quality measured?
 - Work-product is measured by determining how accurate the proximity sensor is with detecting a touch on the keyboard key. In other words, is a touch registered as soon as someone's finger is on the key and not when their finger is above the key. The accuracy should be 95%.
 - The speed of the when a touch is detected by the software after the physical touch was made needs to be less than 70ms.
- How many hours does it take to perform the work?

- For the individual with intermediate skills in C++ and the Arduino IDE, it should take them 24 hours to write the software that outputs the capacitive sensor readings for one capacitive key. It should take them 16 hours to modify the code to work for 12 capacitive keys.
- When there are 8 hours left of software work, the individual with intermediate skills in Arduino hardware will begin their work, and it should take them no more than 8 hours to integrate the Nano and MPR121 for 1 keyboard key.
- Next, it should take 8 hours to add the software to the Nano, test, and resolve any bugs.
- The following should occur in parallel:
 - The individual with intermediate skills in C++ and the Arduino IDE will test to find a threshold in capacitance that indicates a touch was made. This will take 8 hours.
 - The individual with intermediate skills in Arduino hardware will also get 8 hours to give capacitive sensing functionality to all 12 keyboard keys.
- 24 hours will be devoted to testing if the threshold accurately detects a touch in multitouch scenarios by the hardware and software individuals.