# Tool

## Components

* Pi zero w
* 3 sensor dev boards
  + DHT11 (Temp/Humidity)
  + SW-420 (Vibration)
  + ADXL345 (accelerometer)
* dc/dc converter
  + With necessary circuitry
* 12 V Battery
* Mechanical switch
* LEDs
  + Enable/Disable
  + Power on
  + 12V
  + 5V
* 1 PCB
  + Hold all above-mentioned components (except pi)

## HW

* A PCB will be designed to fit on top of the pi zero w. This PCB will dc/dc converter (along with its necessary circuitry) that will step down the voltage coming from the battery to 5V. This 5V will be used to power everything component of the tool.
* The PCB will have jacks for all three sensors.
* The PCB will have a mechanical switch and LED. When the switch is closed, the LED will turn on, and when the switch is closed, the LED will turn off.

## Functionality

* Communications will be established between tool and transponder
* Sensor data will be tagged with date/time, and name of sensor from which it came. Data will be sent to transponder.
* When the mechanical switch is closed, the “time of use” timer will start counting. It will stop when the switch is open. This datum (start time – stop time) will be sent to transponder.
* When connection established with transponder, banked data will be sent to transponder and memory wiped. Any information transponder wishes to send to tool, such as enable timer reset value, will be uploaded at this point.

## Nice2Haves

* Multiple tools (3 should be good)

# Transponder

## Components

* Pi zero w
* 12 V Battery
* LEDs
  + Enable/Disable
  + Power on
  + 12V
  + 5V
* 1 PCB

## HW

* PCB designed for tool will be reused here however, no sensors will be installed

## Functionality

* Communications will be established between tool and transponder
* Communications will be established between hub and transponder
* Value stored in enable timer will be set by transponder
* If enable timer is depleted, then transponder will shut off tool.
* Ability to drop hub to transponder communications to simulate the transponder operating independently from the hub
* When transponder and hub are **dis**connected, the sensor data received from tool will be stored in memory. Once communications between the transponder and hub have recommenced, the stored data will be sent to the hub and then wiped locally.
* When transponder and hub are connected, the stored sensor data received from tool will be sent to hub. The now old sensor data inside the transponder will be wiped.
* The hub will set the enable timer in the transponder

## Nice2Haves

* Multiple transponders (2)

# Hub

## Components

* Pi 4
* Mechanical switch

## HW

* A hub to transponder comms enable switch

## Functionality

* Communications will be established between hub and transponder
* Communications will be established between hub and AWS
* Route instructions from AWS to transponder (transponder timer value)
* Route sensor data from transponder to AWS
* Ability to drop hub to transponder communications to simulate the transponder operating independently from the hub

## Nice2Haves

* Instruction queue

# AWS

## Components

## HW

## Functionality

* Set enable timer in transponder

## Nice2Haves

* With multiple tools
  + Assign a tool to transponder
  + Enable/disable tools
    - Set enable timer to 0
  + Organize data by tool
* With multiple transponders
  + Enable/disable transponders
  + Organize data by transponder