Calibrate accelerometer from scratch

12 December 2025 12:39

Step 1: calibrate average z
- Set one_gz = 1.0 and offset_z = 0.0
- Upload code to acc ---->
- Now acc screen displays measured voltage
  - Record flat, perpendicular and upside down voltage
  - For Battery powered acc this is:
    - up_v = -10.72
    - Mid_v = -0.64
    - Down_v = 8.94
- Set one_gz as the average of -up_v and Down_v
  - This scale the difference on up_v and down_v to be 2
- Offset_z is the difference between the new value (v/one_gz) and the desited value (-1,0,1) times by one_gz
  - For battery acc:
    - One_gz = 9.83
    - Offset_v = -0.09*one_gz
- Delete build folder in pico-examples
- Compile .uf2
- Upload to acc

Step 2: calibrate peak z
- Using swift shaker measure pk_z output at amplitudes of 0-150mv at 50Hz
  - Peak to peak voltage measured using LVDT
    - Amp_mv = Vpp/2
    - Mv-mm conversion calculated by measuring a number of voltages around zero as LVDT is measured on traveling microscope (348.27 mv/mm)
    - Peak acceleration = (amp_mm*(w**2))/g
  - This shaker is used as it produces a more uniform sin wave
- Plot accelerometer acceleration vs calculated acceleration
- Gradient of this line is measures_scale factor
- Alter this number in code and reupload again
- This final calibration is then tested again on swift's shaker before running a DC to acceleration calibration on each shaker and storing this on the z drive

Uploading code to accelerometers
Activate Accelerometer env

In Pico-examples:

1. mkdir build
2. cd build
3. conda activate accelerometer
4. cmake -G "MinGW Makefiles" ..
5. cd accelerometer
6. make

These steps will produce a main.uf2 in build > RP2040-LCD-1.28

Plug the RP2040... into the computer with a cable but hold the boot button on the back at the same time. This should then mount like a usb drive and show up in file explorer.

Drag and drop the uf2 file onto the device and it should automatically reboot and start running the code.