# High level design

Loop through memory, reading addresses that are spaced apart. Once and invalid read is

returned, start to scale back by individual words until read is once again valid. This valid address is the highest point in memory i.e. it is the size of the memory

...

The thought here is a normal step will take a long time to transverse all of memory; take big steps to speed through the bulk of memory, then rewind once we are OOB

# Risks

* Memory address is corrupt, which will return at an invalid address
  + Try to read after it
* Memory controller is bad, and won’t return the invalid address code
  + Unlikely to happen?
* Noise returns valid code
  + Write, then read
* Memory write overwrites bootloader, bricking the board.
  + Find the bootloader size and address, and avoid writing to this space
* Memory write overwrites program, killing the program.
  + Find the program size and address, and avoid writing to this space

# References

https://www.arduino.cc/en/Tutorial/Memory