Did you know that, according to the World Health Organization, e-waste is one of the fastest-growing streams of solid waste on the planet? Here at home, the United Nations Institute for Training and Research even calls it our fastest domestic waste problem, driven by our nonstop craving for new gadgets, ever shorter device lifespans, and almost no easy repair options.

Think about your smartphone: in 2022 the ‘small IT & telecom’ category—phones, laptops, routers and the like—weighed in at 4.6 million tonnes, and roughly 5.3 billion phones alone hit end-of-life that year. Yet despite those staggering numbers, we still face roadblocks: products designed to be replaced rather than fixed, patchy recycling programs, and frankly, not enough of us know where or how to drop old electronics for proper processing.

I’ve always been fascinated by robotics and irony struck me when I realized: the very machines that can pollute our planet also have the power to protect it. It’s kind of poetic in a way. So, I outfitted this prototype that I built to reflect that exact sentiment and if you look closely, you can see that this prototype is off to bury parts that might have even been used to prototype/build them.

Hexapod waste-collection robots are currently being tested in primarily the university setting. Albeit not at mass and funny enough I did not know this when I was designing it. I guess it’s really hard to have an original thought…now I divert to a list of pros and cons of this prototype and hexapods in general….

So yes—maybe I didn’t invent the concept, but I did build this one with intention: to show that even a student project, made of scavenged parts and scrap code, can reflect a larger truth—that robotics isn’t just about automation; it’s about choice.

The same hands that build machines that pollute can build ones that protect.

And this hexapod? It’s not just a robot. It’s a statement. A reminder that innovation doesn’t always mean reinventing the wheel—sometimes it means repurposing the legs.

Thank you.