Open Source Factory Automation

We make robot construction kits like CUBELETS and MOSS, and we make them in our own little factory in Boulder, Colorado. People think that this is bat-shit crazy, but it's not. We don't manufacture our robots here because we're particularly patriotic or to set an example of altruism, we do it because it makes better sense for us from a business (and particularly a financial) perspective.

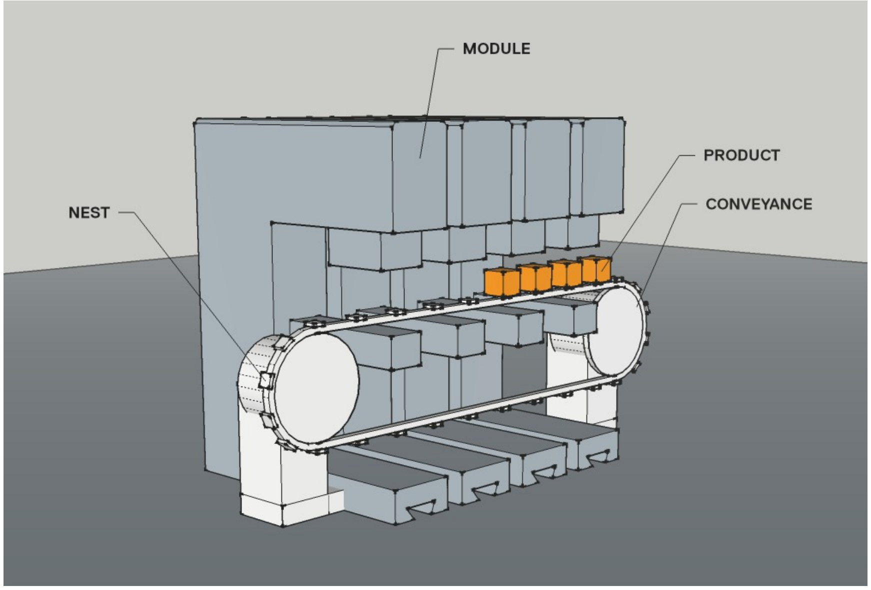
Oddly, the decision to manufacture our products ourselves began on a flight home from China where we had visited five contract manufacturers to talk about making our products for us. I was filled with unease on that flight home; contract manufacturing just seemed wrong for our little robots. As I reflected on the trip, I realized that there were huge differences between the first- and second-tier factories that we visited. At the low-end factories, there were people everywhere: three people running each injection moulding machine to press buttons, inspect parts, etc. But at the high end factories, there was nobody on the assembly floor. There was only the rhythmic buzz, click, whir of gantries and robot arms doing all the work automatically. Wait a minute, if the best manufacturers robotize their entire operations due to Chinese labor getting so expensive in the last few years, maybe throwing in with an automated manufacturer would be shortsighted. Anyway, our company makes robot construction kits so if anyone is set up to robotize and automate a factory, it's us!

Typically, when you automate a factory, you'll pick up a few million-dollar KUKA or ABB robot arms or a Baxter robot that makes smiley faces at you. These approaches entail a huge financial investment, and they're way overkill for the work that we do. Our commercial products are tiny, simple robots, and many of the assembly tasks involved in their construction are tiny, simple tasks. We built some more tiny, simple robots to automate repetitive tasks and save time and money on our factory floor called FARKUS. FARKUS is the open source Factory Automation Robotics Kit for US! That's "us" like in *you and me and all of us*, not like in *United States*, by the way.

FARKUS is more than an assembly robot, it's an ecosystem of simple robots that work together. In the same way that our commercial products are made out of lot of tiny robots, the power of FARKUS lies in the modularity and interconnection between various modules that can be reconfigured quickly to support different assembly processes and production lines.

At a basic level, a factory performs a bunch of operations to a bunch of products. The Modular Robotics factory, for instance, uses operations like soldering, ultrasonic welding, programming, testing, and printing to make products like Cubelets and MOSS. FARKUS consists of several parts: *movers*, to move products through the factory; a collection of *modules* that perform operations to the products; a *control* system to manage the ecosystem; and *nests* to cradle the products as they make their way through a production line.

We've designed and built two movers so far. The Conveyance is something like a conveyor belt: it uses a stepper motor, a Geneva mechanism, and chain drive to move products along a linear path through a series of modules. The Crousel looks more like an old slide projector and moves a carousel of a few hundred products (we're using it for small printed circuit boards) around so that each of them can be programmed/tested/operated on by a module. Other great movers for future design will be a vibratory bowl feeder and a mechanism to separate passed and failed products from the end of a conveyance.

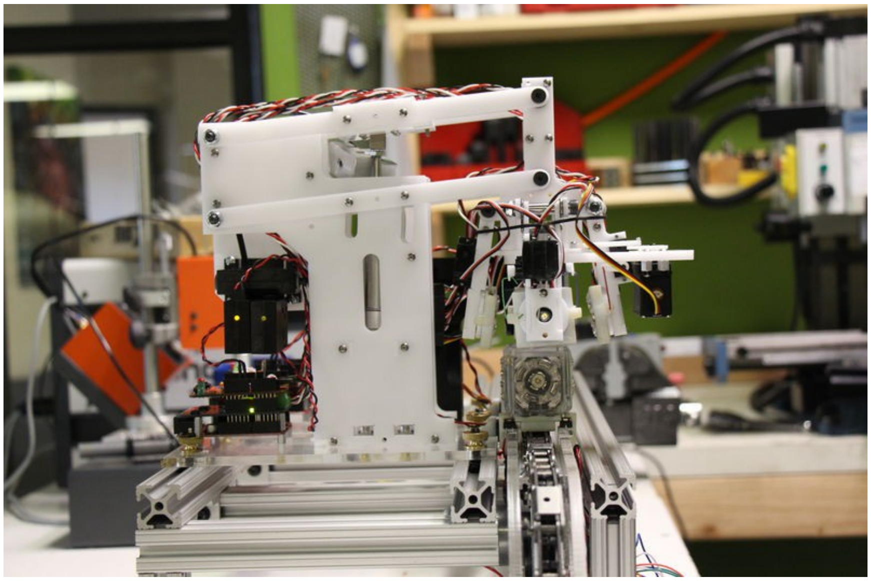


Caption: A diagram of a FARKUS conveyance with four attached modules.

Modules are easily swappable on a conveyance or a crousel so that FARKUS systems can be quickly customized. On a typical conveyance, say, a Cubelets Flashlight Tester, we'll have four modules. The first has a mechanical arm that tests all five faces of the Cubelet for mechanical defects: short circuits or lack of connectivity. The second uses an Arduino to program the Cubelet's firmware, the third has another mechanical arm to simulate five connected Cubelets and make sure the data transfer is correct. And the fourth arm has a photocell to test that the actual flashlight LED on the Cubelet is illuminating correctly at various states. As a Cubelet makes its way through the series of modules, the Control module keeps track of its progress and state, and there's room on the Conveyance for a fifth module that will put Cubelets that pass the tests in one box and those that fail in another. We've built modules to test and program circuit boards, to test for short circuits and connectivity on Cubelets, to ultrasonic weld plastic bits, to test digital communication, and to functionally test robot drive mechanisms.

Another thing you'll notice at a traditional robotized factory is a set of little white boxes called PLCs (Programmable Logic Controllers). PLCs control the robot arms and conveyor belts. They are simple and reliable. However, the cost is about $2000, and they consist only of a circuit board and a few relays. We use $20 Arduinos instead. I'm not suggesting that PLCs are always a bad idea: In some factories, giant robot arms swing steel beams around and could kill someone after an electrical glitch. Other factories can lose millions of dollars per hour if the production line stops running, so the reliability of a PLC is necessary. In our little factory, though, simple, safe, tiny robots with large red STOP buttons can do the work. The agility that this brings us means that when faced with an outage, we can easily reconfigure the robot ecosystem or begin building material buffers in other parts of the factory.

A custom Python application called FARKUS Desk controls everything. Desk currently runs on a Raspberry Pi and uses straightforward serial data transfer over USB to communicate with the modules (each contains an Arduino), movers, and individual products. Desk conducts the orchestra: it controls the operations at each point in the system, and sets and keeps track of state of each part. We're currently using a touchscreen display to interact with each FARKUS Desk, but it'd be nice to have everything connected to a server so that factory status could be monitored and tracked from anywhere.



Caption: A FARKUS module that tests digital communication on six faces at once about to descend onto a Cubelet during final testing.

The last major component of FARKUS is a set of nests: electromechanical connectors that attach to the products themselves as they make their way down the production line. Both Cubelets and MOSS have simple, self-aligning magnetic connectors, so building their corresponding nests only entails a quick 3D print of a connector and a PCB to break out the electrical connections. Many products won't need electrical connections -- if your FARKUS system manufactures T-shirts or laser-etches Jesus faces into toast, you won't need serial data to and from the toast. Nests for circuit boards are usually laser-cut plastic sandwiches inspired by the Sparkfun manual pogobed design and fitted with header pins to interface with a Crousel. We also have nests that are milled from steel so that MOSS faces can be ultrasonic-welded into their plastic frames. Depending on what you're making, designing appropriate nests for a product can be anywhere from straightforward to extremely complicated, and this sort of necessary customization is one of the main reasons we decided to open source FARKUS.

I'm not filled with Open Source dogma; I don't think that everything should be open source. FARKUS is open source even though our consumer products are closed source. Our consumer products are designed to be used, not to be taken apart and rebuilt. Hacking into Cubelets and circumventing the battery protection circuitry, for instance, could cause the product to subsequently catch fire in a nine-year-old's hands. But open source is perfect for FARKUS for a couple of reasons. Automating a factory requires a significant amount of customization. Although FARKUS can provide common features like conveyor modules, material handling modules, indexing, testing, and sorting modules, an actual implementation will need custom tools, jigs and nests that handle the actual product being manufactured perfectly. Factories are usually run by capable engineers who can modify and hack assemblies safely and at their own risk. We also decided that we didn't want to get into the factory automation business.

FARKUS could have become an in-house, proprietary system, but we're thrilled to think about a future in which products aren't all made halfway around the world. FARKUS is still an experiment, and we're publishing the results of each build: both time saved and return on investment. We're amped on local manufacturing and I heartily encourage you to build a conveyance and a couple modules, share them, and manufacture a few thousand widgets right in your garage.

FARKUS

Details, CAD, code, schematics, videos, and more info at: www.modrobotics.com/farkus