V03.7

Introduction to Handibot Software and Handibot Apps

I. Hello Handibot!

If you've seen Handibot, you'll appreciate it's capable of doing some amazing cutting, drilling, carving, and machining … all in a nifty little, portable, power-tool package.

However, it's going to take innovation and creativity to fully enable this new, "Smart Tool" format. Software apps and accessories will come to power a wide range of jobs that Handibots and other Smart Tools are going to take on. To grow this future of digital power tools, there are opportunities for makers with ideas, developers who can help make the ideas happen, and probably plenty of people who can do both.

We'd like to encourage you to jump into the Handibot development ecosystem. This document outlines the elements of putting Handibots to use: first, specifying the physical/mechanical capabilities of the tool; then, describing the new "FabMo" software platform that runs the tools. The FabMo platform carries out "jobs" and tasks that will be facilitated with Handibot Apps, and we lay out the basic concepts for these Apps here. Whether you just want to contribute some thoughts on how to use the tool, or want to code a cool Handibot App, or to design an accessory – we look forward to your getting involved.

Handibot Capabilities

First and foremost, Handibots are a new category of digital power tool. They are about bringing powerful capabilities of precise and controlled cutting "to hand" – to extend an individual's physical creativity and productivity with digital technology – all in an engaging portable format.

In developing Handibot, we made few compromises because we wanted to give this tool the widest possible range of future uses. Handibots have both power and precision. We've packed a lot of potential into a little box ...

- A Handibot can cut, drill, carve, or machine in a 150mm x200mm x100mm (6" x8" x4") envelope; the tool works down through its base to cut material underneath; with a Handibot you *take the tool to the material* you are working on rather than putting material into a tool or onto a fixed table. The material will typically be held in place during cutting by pressure from the weight of the tool itself.

- In its basic mechanics, a Handibot tool does 3-axis, XYZ motion; but the onboard controller has 6-axis capabilities to enable attachments and accessories that extend the kind of work that can be done; there are also up to 12 digital inputs and outputs and several analog channels that are potentially available for added accessories and features; the onboard single-board-computer (SBC) supports several communications options. There's lots of potential functionality -- think robotics …
- Fitted with a 1.25hp router; Handibot's force and rigidity allow it to work in wood, plastic, aluminum, foam, and other similar materials; yet high accuracy [0.006mm (0.00025") step resolution] and rigidity allows fine machining for work such as jewelry or printed circuit boards (PCBs).
- Fitted with a drag-knife instead of a router bit, Handibots can cut paper, vinyl, cardboard, or other thin materials; fitted with a diamond drag bit they can etch/engrave/mark glass and metal; fitted with a marker, they can draw. What else should we fit them with?
- An "Accessory Base" is available as an enclosed work-holder for blocks of material. The Accessory Base also serves as a mounting bed for a rotary-indexer, 4th axis, or a manual 3+2, 5-axis fixture.
- Handibots are portable but they also function well as small, desktop CNC mills (having better specs than most small mills). To get you started with CNC work, Handibots come with CAD/CAM tools: with V-Carve Pro software that provides easy-to-learn CAD/CAM design capabilities; and, with a year's license to Autodesk's Fusion 360 for more advanced 3-D-model-to-CAM-to-toolpath workflow. Both systems helpfully provide CAD to CAM processing within a single software package.
- We are pleased with this foundational functionality of Handibots as capable, small CNC tools. But our more ambitious goal is to arm these tools with software Apps and hardware Accessories to become job-oriented "Smart Tools" – making any sort of project a whole lot easier – enabling them to do digital making without the need for the sometimes challenging and distracting steps of traditional CAD > CAM.

An Early Handibot Prototype

Handibot as "OPEN" Innovation

Handibot is an "Open Innovation" project. It is open-source hardware and runs open-source software. The digital models for the current Handibots are available on the Handibot website and are detailed in our GitHub repository (https://github.com/Handibot); you can access the software platform work from the FabMo webpage (http://gofabmo.org). By "open innovation" we further mean Handibot will be a both a project and a manufactured product whose evolving hardware design and software are openly shared with a community of users, collaborators, and developers. This approach is a forthright experiment on our part – an expression of our interest and enthusiasm for moving towards more rational, environmentally and technologically appropriate approaches to product development and manufacturing.

We support “open-source” as a methodology because we believe it is an effective way for small companies, small groups, and individual entrepreneurs to leverage collaborative power in order to innovate and remain competitive and sustainable. We believe small companies and distributed-manufacturing using digital-fabrication technologies offer one of the most attractive scenarios for our common future … and that’s why we’re committed to evolving Handibot in this manner.

II. Handibots Run on FabMo

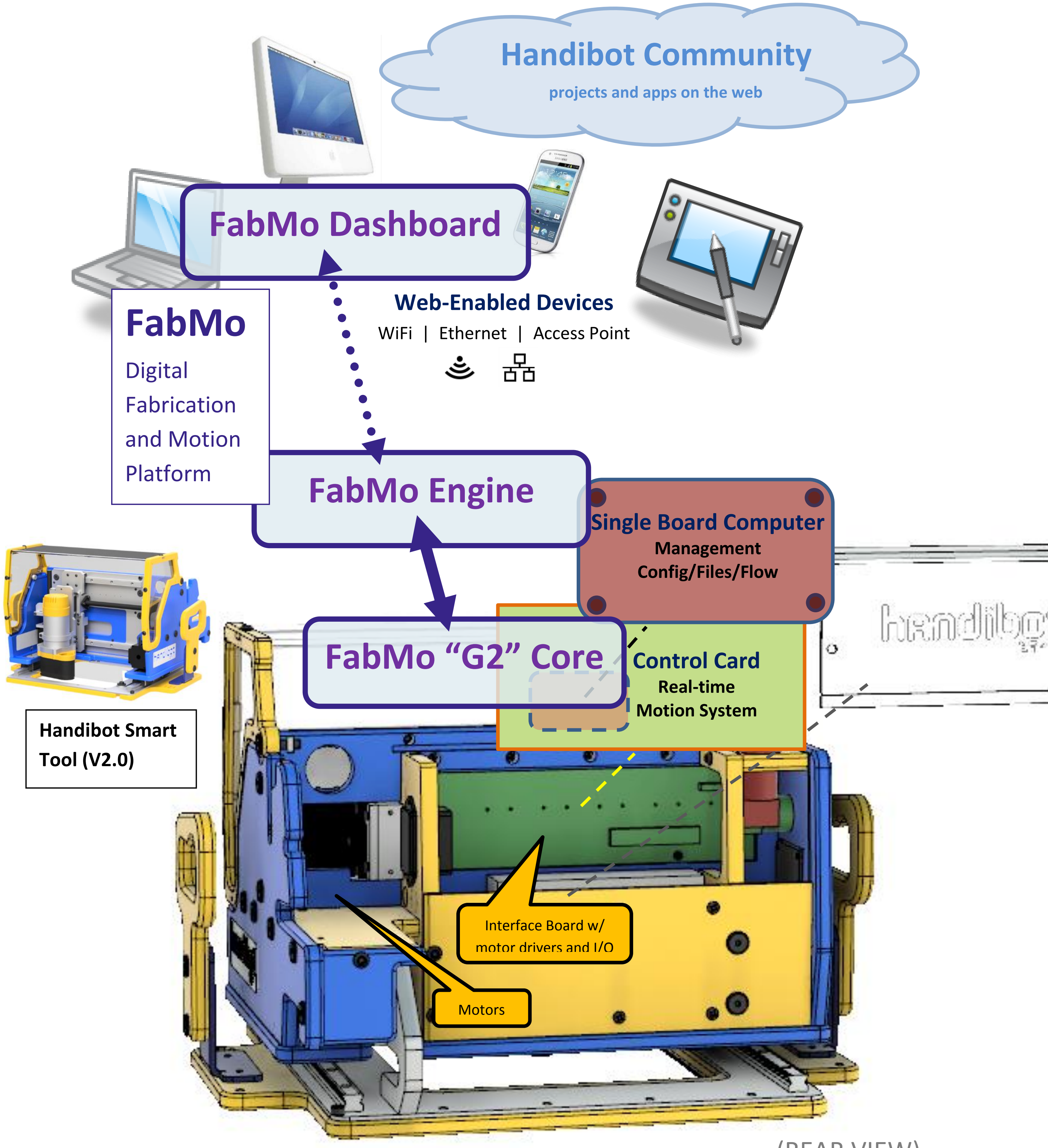
Handibot is the first digital tool to run on the new software platform, **FabMo** – short for digital fabrication and motion (gofabmo.org). By taking advantage of progress in microprocessors and microcontrollers, FabMo offers new, low-cost options for managing and driving digital fabrication equipment, both subtractive and additive. It unlocks equipment from the constraints of specific operating systems and communications methods. It is the perfect platform for Handibot.

Originated by the Handibot team at ShopBot Tools, FabMo builds on a powerful core motion system, called G2, developed by Synthetos (github.com/synthetos). FabMo is open-source software. It is designed to run on standard, electronic hardware components. It is available to be used and adapted by anyone; to run any digital tool imaginable, today and tomorrow. Beyond Handibot, we hope FabMo will come to support a wide range of affordable digital fabrication equipment produced by many OEM’s, as well as run all sorts of custom tool and robotics projects.

FabMo Description

The FabMo platform lives on several hardware components within and outside a digital fab tool. FabMo looks out to the world from a single-board-computer (SBC) residing on the digital fab tool – for us that means it looks out from the Handibot. The **FabMo Engine** serves up the **FabMo Dashboard** to a browser on almost any type of web-enabled device via wireless or wired routes. The Dashboard provides a friendly management and app environment while the underlying FabMo Engine handles configurations, and monitors status. More importantly, the Engine runs jobs and handles files. The Engine supports multiple motion languages, and efficiently feeds commands and streams data to the **FabMo Motion Core** (G2). The Motion Core runs in real time on a dedicated microcontroller. It is an isolated, low-level, real-time, high-performance motion system producing nimble and responsive movement. On a Handibot, the Motion Core is located on its own Control Card which plugs into a CNC Interface board that communicates to the tool’s drivers and motors.

V03.7



Handibot Community

projects and apps on the web

FabMo Dashboard

Web-Enabled Devices

WiFi | Ethernet | Access Point

FabMo

Digital Fabrication and Motion Platform

FabMo Engine

Single Board Computer

Management

Config/Files/Flow

FabMo “G2” Core

Control Card

Real-time Motion System

Handibot Smart Tool (V2.0)

Interface Board w/ motor drivers and I/O

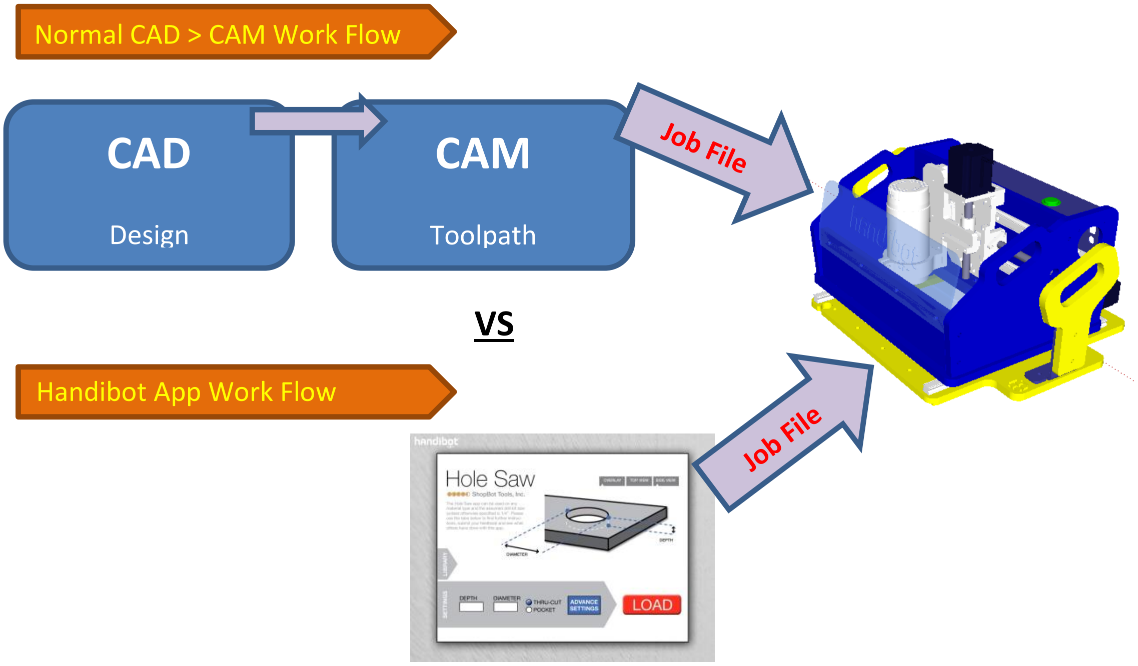
Motors

(REAR VIEW)

Work is done on a Handibot or other digital tool by sending a file of instructions to the tool. The file describes the motions that the tool will then move through to do machining or build a part. Such files are variously called part-files, toolpath files, CNC file, or cutting files. They are primarily the list of XYZ coordinates defining the path along which the tool will move in doing the work. The file may also contain instructions for controlling or modifying other functionality of the Handibot, such as changing speeds or turning on or off output switches. This file is the same type of file as would be created in a normal CAD > CAM > CNC Tool workflow.

We call toolpath files, “job” files. On your Handibot, job files are handled and work is done by the Job Manager. It will carry out jobs that are accessible from a device linked to your tool. You can run a job file that is stored on your tool or on your device; or run a job file that your device can access on a network, or from the web. The Job Manager is always accessible a way to things done.

“Running a job” with the Jog Manager represents the normal CAD-to-CAM-to-digital-tool workflow. For example, you might design a part in a CAD program on your PC; then send the design file to a CAM program; the CAM program would help you generate a toolpath or job file; that you would then send to the Job Manager for execution.



Normal CAD > CAM Work Flow

CAD

Design

CAM

Toolpath

Job File

VS

Handibot App Work Flow

Job File

Beyond this traditional work flow, Handibot’s FabMo platform offers a new approach for using smart tools. It supports “apps” that can more easily access to the capabilities of digital equipment. Apps can do a wide range of common tasks and also help in producing more complex projects. Handibot Apps will make it a whole lot easier to put digital tools to work.

For the Handibot tool user, this new type of app will become a primary work-horse. For many tool-use needs Apps will replace the difficult work-flow of creating and drawing a shape in a CAD program and then using a CAM program to generate a tool-path for it. Having to master both CAD and CAM makes it hard for many people to put CNC to use. Needing to do the work of CAD-to-CAM also makes using a CNC inefficient for routine tasks.

A Handibot App will create the tool-path file to send to the Handibot – replacing the CAD/CAM process. The Apps' functionality for the user will generally be to make the process of defining the machining for a regular job EASIER – easier than engaging in the full CAD/CAM process … and more like "click to print".

III. What Kind of Apps Will Handibots Have?

Apps will usually be more than "hard-coded", fixed cuts. In most cases, we expect apps will be "parametric". That is, the app will turn a generic need into the specific instructions required for a user's particular situation after receiving some input. For example, a Handibotter might want to cut a specific size hole in a board – say 4.37" diameter at a depth of 0.52". A "Hole-Saw App" would allow the hole to be easily defined parametrically at the time of need and the cut-out quickly and precisely made – nearly push-button.

In order to encourage app ideas for development, we've created a Handibot Community (https://handibot.com/community.php) site to collect creative suggestions for Handibot uses. A number of suggestions and ideas have already been contributed to this interactive site. We're looking for ways to further encourage and reward these contributions, but you can already find a lot of really interesting proposals on the site to stimulate your thinking about ways to put smart tools to use. Here are just a few:

Basic Apps might include …

- Cut variable size holes, at designated locations, with countersinking, pocketing, etc..
- Cut-off functions, diagonals, joints; lots of sizes, and varied types of materials
- Cut-outs for fittings such as lock-sets and hinges; matched to products via libraries?
- Cut-outs for fittings such as electrical or lighting items or plumbing

Apps that work with ACCESSORIES …

- Functionality can be enhanced by developing registration systems that allow positioning and prompting for "tiling" projects larger than the Handibot's base work area; such systems might be simple and use manual registration, such as fences and fixtures; functionality might be further extended to automated motion on tracks, rails, or wheels for re-positioning. We've already created one jigging system for manual registration which is available for download and cutting or, for purchase from us pre-cut (Large Material Jig). But there is plenty of room for other approaches to tiling files using Apps. (Note: the V-Carve Pro software provide general purpose tiling functionality.)
- Apps might take advantage of a 4th axis for rotary indexing; extended length could be added by supporting registration down the rotating part. (See: Accessory Base; Rotary Axis; Manual +2)
- Vertical work of many of the basic types could be facilitated with support systems such as stands, bracketing, or clamps for positioning against walls or other surfaces (there is an example on our Kickstarter project site).

Apps that work with <u>ADVANCED</u> ACCESSORIES …

- Integration might be done with a vision system for positioning the tool (e.g. Kinect)
- An image system could also be used for digitizing shapes in order to do accurate, on the job-site, fitting of trim or components (e.g. edge trim along a stone fireplace)

Construction Job Site App Ideas …

- Pocketer (cut pockets, half-laps, etc)
- Beveler (make beveled cuts…as done with compound miter saw)
- Curver (cut curves/splines)
- Engraver (v-carve text or clip art into material for labeling or signage)
- Puzzler/Joint-Maker (cut any of a wide variety of joints to connect boards or parts)
- General chop-saw applications; especially where distances or angle are critical and change frequently; fit to a job-site type chop-saw stand; could be enhanced with fixturing or automation for handling piece length
- Door frame joinery and cutouts
- Inlays, counter and floor treatments
- Arches circles and ellipses with some sort of registration/indexing; vision system input?
- The company, Homebuilt (http://www.homebuiltcompany.com), proposes to use Handibots to do onsite fashioning of joints and extra parts for digital home fabrication.

At Home Apps and Projects …

- Construction uses as above but also more general hobby crafting; portability creates opportunities that are not possible with full size, immovable, CNC systems; yet, registration allows working in a much larger area than the tool's base work area.

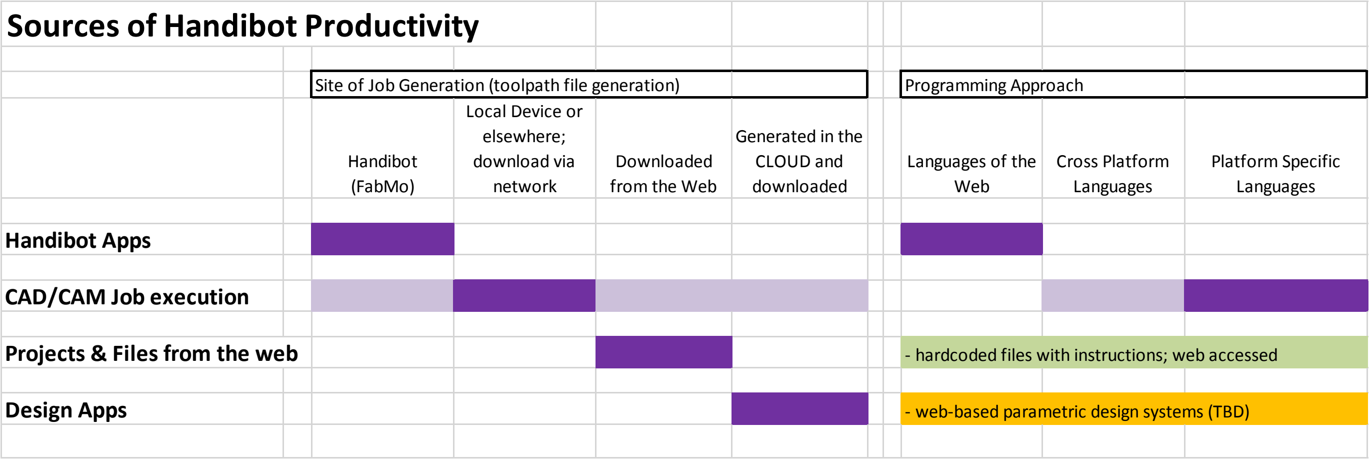
School Apps and Projects …

- A Handibot is a tiny desktop CNC for schools and students of many types … at a school-friendly price; we are getting lots of suggestions for student projects. Apps oriented to education would offer an easier introduction to digital manufacturing that having to first master 3d modeling and CAD.
- From models and prototypes to full size structures, theater tech, museums, and performance arts programs are ready to put interesting apps to work.

Further app concepts, examples, and discussion can be found on the Handibot blog (http://handibot.com/blog).

A Broader Perspective on Digital Fab Work, Jobs, and Apps

We've described some of the exciting possibilities for "Handibot Apps". But let's put them in the context of all the types of jobs that smart tools will carry out and how we think about the characteristics of this other work. We've outlined 4 categories of work for Handibots. They have different purposes, work in somewhat different ways, and are likely to operate from different locations or the ecosystem:



Sources of Handibot Productivity

	Site of Job Generation (toolpath file generation)				Programming Approach		
	Handibot (FabMo)	Local Device or elsewhere; download via network	Downloaded from the Web	Generated in the CLOUD and downloaded	Languages of the Web	Cross Platform Languages	Platform Specific Languages
Handibot Apps	■				■		
CAD/CAM Job execution	▨	■	▨	▨		▨	■
Projects & Files from the web			■		- hardcoded files with instructions; web accessed		
Design Apps				■	- web-based parametric design systems (TBD)		

1- **Handibot Apps** as we've described are onboard Apps that carryout job functions such as cutting a hole of a specific size, making a joint of a given size or type, doing some sort of simple marking or lettering, and so forth – doing a basic "job" resulting from the user making a few choices or setting a few sizes. This type of Handibot App might be considered a defining type for on-the-job "smart" functionality. Because Handibot Apps are fundamental 'tools' for using a Handibot, they will typically reside on the tool (that is, in the FabMo Dashboard on the SBC). Being resident means they will be available anytime the tool is in use, always ready from whatever device that user is accessing the tool with and whether or not there is an internet connection available in the work environment.

2- **CAD/CAM Job execution** is the traditional way to use digital fab equipment. We expect it to get easier, but it will continue to be well suited to carrying out a particular user's complex, designed projects. At the moment, CAD>CAM work is best done within software on a user's device – typically a PC. New options are becoming available to download shared projects from the web or to do work from the cloud and download the job to the tool.

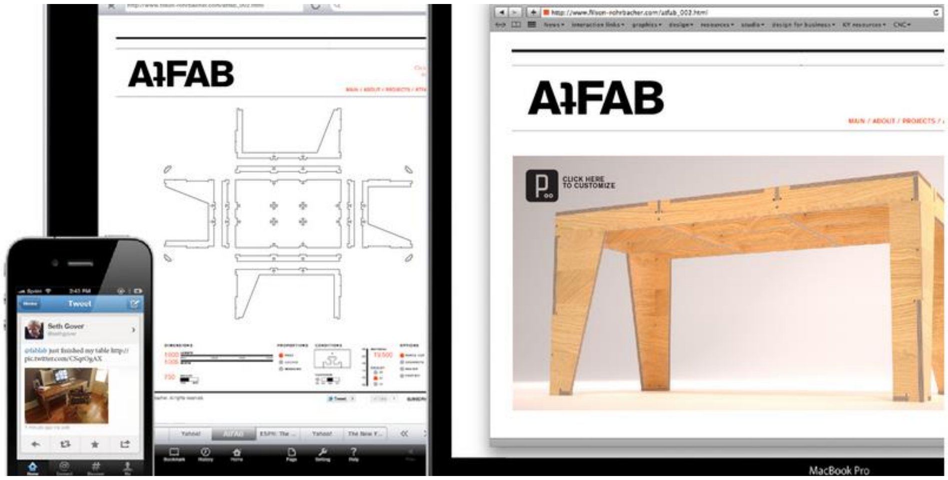
3- **Projects & Files from the web** are fixed, predefined, and complex cuts for a unique project. These can be made available by someone who has developed a project in a CAD/CAM program and wishes to share the project. It could be anything from a musical instrument to a boat. The primary aspect of a "Project" is that it would be a toolpath file, or set of toolpath files and instructions, that do not involve much user interaction or input other than making a selection from a list, gallery, or catalogue. The distinction here is that these files would typically be used once and would not be considered part of the regular functionality of the tool. One could imagine that Projects might have a few options and selections that are available at the time of download or a cutting time. As well, runtime modifications would allow for some scaling of project files. But all of this customizability would be limited because the projects are fundamentally based on fixed or hardcoded toolpaths. We expect to have a wide range of Projects available on the Handibot Community website. A subset of this category would be libraries of predefined cut-outs for items such as sink fixtures, light mountings, or lock-set holes provided by individual vendors. Job files from Projects on the web are likely to be used only a single time and there is thus little need to save or process them on the Handibot or user's device.

4- **Design Apps** are ambitious apps that involve creating a design or project rather than handling single cutting or machining operation. Design Apps are the "parametric" concept applied to much more complex designs. These Apps fulling integrate input from the user in customizing the items to be produced. That is, they will be broadly "parametric" – having user adjustable values or parameters for sizes, features, and construction. They might help a user create a jewelry item, a piece of furniture, a house!

A Design App might allow the user to define the height and width of the table, as well of the thickness of the material used to make it. These parameters would be used to modify the underlying coffee table design and output the specific tool-path to make the exact size coffee table that the user wants, with parts adjusted for the material that will be cut. This customization can be more complex than simply changing the scale of parts. It might involve things like the shape, size, and placement of joints – even the adjustment of joints to fit the thickness of the material. The App developer would have done all the heavy lifting to define the generation of the toolpaths from the specified parameters; with the user having just defined the specifics of the design as it relates to their own needs.

Of course, the App might have a very compelling interface that allows the user to preview what the table will look like, maybe with a couple of sliders to manipulate length and width. For example, have a look at the sophisticated chair-making-system being developed by our friends at SketchChair (http://sketchchair.cc/) -- this level of complexity goes beyond what would routinely be used as a Handibot App, but it conveys the kind of work that could be produced with more ambitious "Design Apps" for Smart Tools.

Another example of Design Apps is the work of atFAB (http://atfab.co/) who have designed several distinctive, contemporary furniture items and are exploring systems for making them parametric (http://makezine.com/projects/make-38-cameras-and-av/cnc-maker-bench/).



@fablab just finished my table http:// pic.twitter.com/CSqrOgAX

For developers, Design Apps thus involve creating a design and determining how its components will be parameterized or customized – as well as presenting it to the user and generating the code from the design that is used to run the Handibot. In the best of all worlds, development of these types of Apps should be less about the coding tasks and more about design.

We are looking forward to the evolution of "designer friendly" systems which will offer an environment for design-oriented developers to create parametric designs and make them ready for delivery as Apps – with the presentation and tool-path coding being handled behind the scenes. Imagine a furniture designer, for example, who wants to create exciting customizable furniture but who is not a programmer or interested in the creation of the App interface itself. That designer would like to be able to utilize some type of parametric design tool to help create designs for fabrication based on methods for arraying parametric components. A number of designers and developers have experimented with parametric apps as web/cloud services, but at this point there is not yet a system for designer to develop such parametric delivery systems that does not involve low level programming.

IV. Get Started Building Handibot Apps

For the moment, here, we are focused on Handibot Apps – the tool-resident type of app that lives on the tool, and is supported by the FabMo system. It is the front line of making digital fab easier and it instantiates many of the general principles of more complex Design Apps. And, now that you've gotten an idea of the range of jobs a Handibot might carry out. Let's consider in more detail how jobs are done with Handibot Apps and begin to look at how to build them.

In brief, a Handibot App is available to a user in Handibot's, FabMo Dashboard. The App, after interacting with the user, will seamlessly pass a Job file to the FabMo Engine on the Handibot, resulting in the Handibot carrying out the Job.

<u>The FabMo API for Apps</u>

A Handibot App is installed in the FabMo Dashboard and it will remain available on the tool unless removed. Apps are coded in the languages of the web – HTML, CSS, and JavaScript. The FabMo Dashboard has an App installation and management system that will load your App making use of a standard packaging framework.

For your App, the FabMo Dashboard handles all basic interaction and tool functions and provides a full set of displays appropriate for most fabrication work. Through the Dashboard, FabMo manages wireless or wired connectivity from the SBC on the tool to the user device. The job-file-processing is done by the underlying FabMo Engine, which carries out all low-level dialogue with the G2 motion core that handles timing, motor action, and I/O. This leaves you free to concentrate on developing an interface for interaction with Handibot users and on generating the Job toolpath and other tool actions.

To accomplish these things the FabMo Dashboard has an accompanying JavaScript library, fabmo.js, with its own API (**LINKS**). These Dashboard API's support your interaction with the Dashboard interface, with system-functions, and with the tool. API functions are provided for:

- Status & Configuration
- Job Management
- Job Execution
- Direct Commands

The FabMo App Developer website (**LINKS**) provides a starting point for developing Handibot Apps using this API. It summarizes how apps are built and work; it provides an App Developer Quick Start Guide; and, most importantly, it links you to a definitive set of example apps, project templates, and other resources.

The "Job File" that Underlies Apps and Other Work

The Job files produced by your app define the motions that the tool will move through to do work. Your App may use either of two different language syntaxes to communicate cutting and machining instructions to a Handibot (FabMo currently supports two motion languages, but is flexible and open, and addition language runtimes can be created within the FabMo architecture):

a. Standard (NIST) g-code files; g-code is a universal syntax for communicating with CNC (computer numerically controlled) tools including 3D printers. There are many variations of g-code, some more ugly than others, but Handibot will work with a basic subset of the very common NIST-defined codes. Virtually all CAM software will output standard g-code.
b. OpenSBP code; this is an open-syntax code for controlling digital tools. ShopBot contributed the code format to the CNC community many years ago. It is similar in concept to g-code but uses symbols, syntax, mnemonics, and programming extensions that are much easier for humans to read and understand than g-code. Many current CAD/CAM software systems will output OpenSBP code as well as g-code. Because OpenSBP adds additional programmability compared to the g-code instruction set, it may make some types of app development easier (www.opensbp.org).

Developers can use either type code for communicating their job files to Handibots. If you are already comfortable with g-code, you should use that. If not, OpenSBP will be easier for you to get going with. We've included OpenSBP support with FabMo because it is a useful alternative to the obscurity of g-code. Additionally, both runtime languages are supported in the FabMo Editor. You can use the Editor to create, edit, and test-run versions of your jobs or portions of your job. It provides a good way to explore the effects and actions of motion commands.

The FabMo Development Environment

The most straightforward way to develop Handibot Apps is with a Handibot. That's because all the hardware is present. You can work between a device (typically a PC running Linux or Windows; working with an editor and the developer functions of a browser such as Chrome) and the Handibot to create, edit, and test your App code.

You may find it more efficient during initial development to run the FabMo Engine on your PC rather than Handibot's SBC. In this case, to fully simulate the interaction with the tool, you will need still need a Handibot Control Card to do low-level interactions with the Engine. You substitute an Arduino Due for the Handibot Control Card if it is loaded with the current version of G2.

For just testing the basic interface of your App, you can just run it in the browser. There are acknowledgements of calls made to FabMo, but there will be no functionality, nor will you get the full effect of operating within the FabMo Dashboard.

It may be helpful to actually see FabMo in action. You can do that by logging into our FabMo server (LINK) that is set up to simulate the operation of an individual Handibot. You'll be able to experience the look and feel of several example apps.

Additional FabMo Platform Information

FabMo has an HTTP and Websocket API. This is how the engine and the dashboard communicate, and how "native" applications might communicate with FabMo. These will be of interest to anyone wishing to communicate with FabMo from outside software (say sending directly from a post-processor to the Job Manager), or anyone just looking for a better understanding of the intrinsic support for the Dashboard.

The FabMo platform itself is an open-source project for which we are seeking participants, developers, and partners. The project already includes several OEM's as well as individual developers. Please visit the FabMo GitHub site (http://gofabmo.org/) for more information.

Where Will I Distribute or Sell My App?

It's up to you …
We intend that Handibots be as open as possible. You will be free to distribute Apps (and run them) however you wish. That includes selling them through your own site or gallery, or just giving them to your friends.

But we hope you will consider …
We do have our own preference. We encourage you to consider distributing/selling your Apps through the Handibot Community website. This site will be the center of Handibot Community activity and a source for Handibot resources for all Handibot users. It will be the natural site for users to come when looking for a Handibot App.

We encourage open-source apps, but we also understand that software development takes time and effort; we believe that effort should be compensated in one way or another. Our store will offer you an automated solution for delivering your Apps and receiving payment for them. It will support a developer gallery to describe and feature your Handibot Apps and to manage reputation ratings so that as the Apps ecosystem develops, users will be able to easily find your Apps and to share their confidence and delight in your work.

Additional Thoughts on Developing for Digital Fab in General …

Our discussion of Apps and Projects here is oriented to Handibots. But, we view these concepts as being appropriate for development of software for all coming Smart Tools and for future digital fabrication tools in general. Thus, though a developer might worry that work on Handibot projects addresses a relatively small audience, we imagine that these types of Apps and Projects will have applicability far beyond just Handibots. Indeed, one reason for our job-file focus in Handibot App development is that for the foreseeable future, CNC-style toolpath files are likely to be the primary way digital fabrication equipment, both subtractive and additive, is run. Your development effort will not be limited to our initial Handibot market – you will be developing Apps that whose output is right for small Smart Tools today, and for extended digital fab markets of the future …