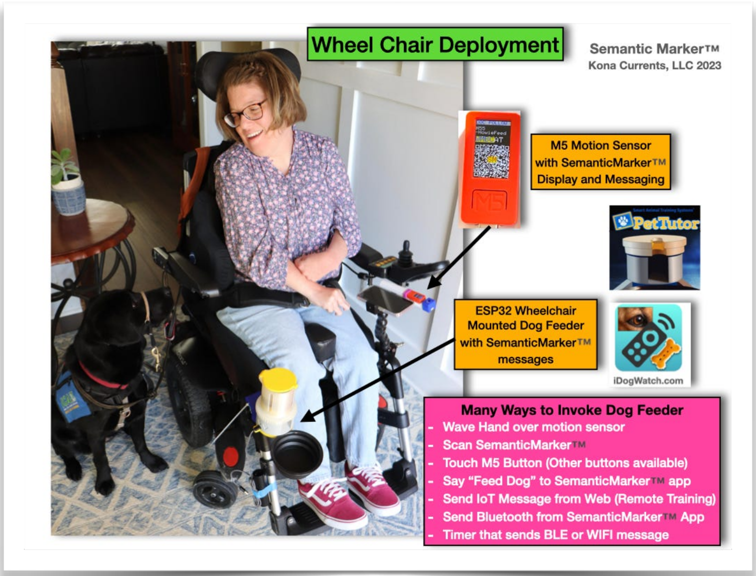
Renewed Agency as #Abilities meets Internet of Things (IoT), Messaging, 3d Printing and Optical Markers

Tuesday, March 12, 2024

Scott Moody, Kona Currents, LLC


Wheel Chair Deployment
Semantic Marker™
Kona Currents, LLC 2023
M5 Motion Sensor with SemanticMarker™ Display and Messaging
PetTutor
ESP32 Wheelchair Mounted Dog Feeder with SemanticMarker™ messages
iDogWatch.com
Many Ways to Invoke Dog Feeder
- Wave Hand over motion sensor
- Scan SemanticMarker™
- Touch M5 Button (Other buttons available)
- Say "Feed Dog" to SemanticMarker™ app
- Send IoT Message from Web (Remote Training)
- Send Bluetooth from SemanticMarker™ App
- Timer that sends BLE or WIFI message

DAUNTING #ABILITIES CHALLENGES ARE MET WITH NEW TECHNOLOGIES (PHOTO BY SMART ANIMAL TRAINING AND SEMANTICMARKER.ORG)

Daily challenges faced by those with disabilities are *daunting*; technologies we take for granted are hard to fathom (*e.g.*, phones, apps, buttons, speech, vision, mobility.) This lack of "agency" — doing it yourself — is frustrating. The result is a large population segment that could use help. There is a new push in a segment nicely called #Abilities — where new technical concepts help users regain "**agency**" through renewed **Abilities**.

21st century technologies can help. The Internet of Things (**IoT**) is a new paradigm with lots of small devices that sense their environment and through messaging, collaborate between users and other IoT devices. Maybe steering a wheelchair by eye tracking, rewarding your service dog via an IoT dog feeder, using motion or vision sensing to perform operations. Smart!

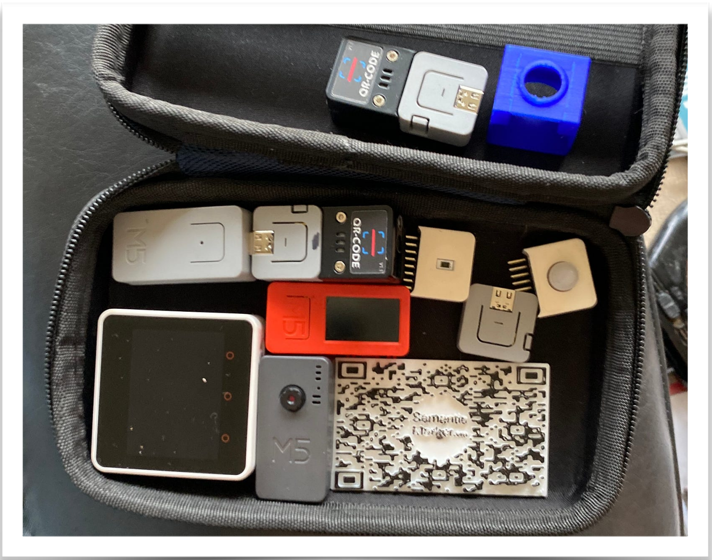
> #Abilities needs the mind set of Trinity (from the Matrix movie)—when asked if she can fly a nearby Helicopter. She responds with “not yet” and calls her remote agent, Tank, to download that flying knowledge. The result is renewed Agency.

INTERNET OF THINGS (IOT) ABOVE THE REMOTE ALASKA ARCTIC CIRCLE TALKING WITH BLUETOOTH WITHOUT WIFI. AUTHORS PHOTO ON LONGEST DAY

Table of Contents

Briefcase full of Things: Internet of Things

This paper will discuss how a *Briefcase full of Things* — the Internet of Things (IoT) — has a unique opportunity to bring back *agency* to the #Abilities market, especially when thinking outside the box (like a 3d printed box.)



BRIEFCASE FULL OF THINGS: INTERNET OF THINGS (IOT) — CONTROLLERS AND SENSORS (FROM AUTHORS COLLECTION)

#Abilities Challenges (touching, selecting, sensing)

The **Abilities** challenges are daunting! Touching buttons whether physical or on a smart phone are hard. You've seen images of Steven Hawkins in a wheel chair using voice commands, others blowing in a straw to perform operations, using their head to bump into a "button", or hand waving for motion sensing. Eye Tracking is a new approach where a camera tracks the users eyes and they can then type entire email messages or steer the wheelchair.

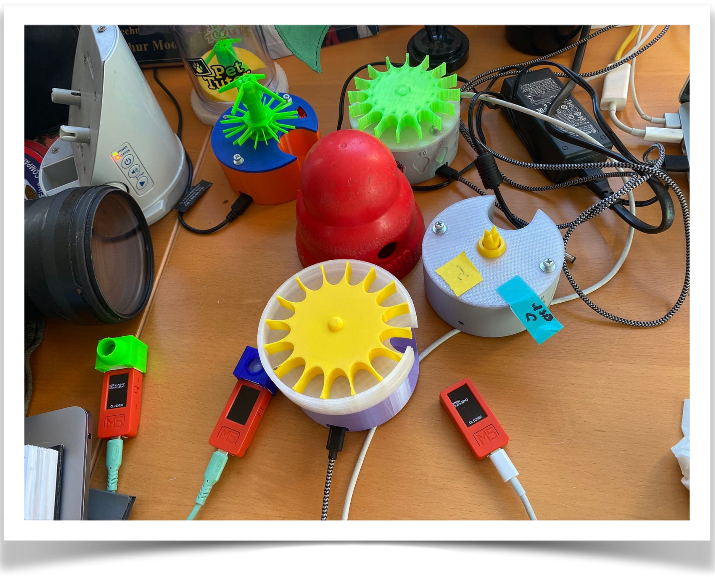
Wheelchairs are a major #Abilities segment but other areas are important. Even non-technical users have trouble just changing the channel on a new TV. Many are using a non 21st century phone (eg. flip phones with big buttons are normal.) Thus “apps” are rarely used (good luck ordering a Uber ride-share.) Any change in routines are hard. The result is frustration and limited *agency*.

> Stroke victim’s have a hard time using the phone, reducing their communication skills and reducing their agency. —Laurence DeShields—ER Doctor

The concept of a remote agent or handler is important — and made possible with this IoT framework. Just like *Trinity’s* remote agent *Tank*, setting up an environment for success is valuable. Traditionally a helper programs all the *things* — creating a routine (like managing the *Uber* ride-share.) But the potential for the user to do this programming themselves is being explored. Thus common tasks can be stored and re-played by the users themselves. Various IoT options help with the re-playing. Even pointing at an optical visual marker can trigger these operations (see **Semantic Marker™** and **PAT** section below.)

A new generation of ESP-32 based Internet of Things (IoT) devices and sensors, when connected through a robust global messaging framework is a powerful foundation supporting #Abilities. Then wrapping the IoT devices in unique 3d printed models — opens up new and unique *agency* potential. Remote control by remote agents is also possible with the messaging framework described below.

ESP-32 IoT Sensors and Controllers



COLLECTION OF ESP-32 IOT DEVICES FROM THE AUTHOR AND SMART ANIMAL TRAINING (E.G., REMOTE IOT DOG FEEDER)

The M5 Stack company is leading in making and packaging a new suite of IoT devices based on the ESP-32 chip and Arduino development environment. 100's of devices, sensors and controllers are available ranging in price from $3 US to $20 and more. Combined with 3d printing (described later) unique packaging and deployment is possible (such as special wheel chair mounts for the dog feeder shown above.)

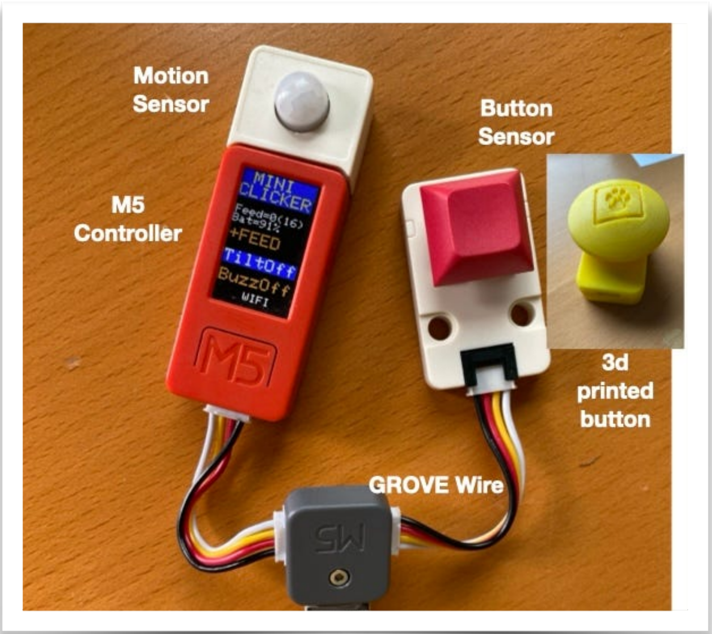
> The ESP-32 chip is a game changer by sharing a radio between Bluetooth and WiFI, providing a dual core, real-time OS and enough memory—in the size of a quarter. This means that every individual IoT device can now work off the grid using Bluetooth Low Energy (BLE)—but when a WiFi network is added, the new potential is for world wide connectivity down to every edge node. Coding commonly uses a mix of C and C++ in a very

high level component architecture (versus traditional low level device coding.)

IoT sensors monitor the environment and report their findings (via controllers) through messages (ie JSON over MQTTor Bluetooth.) They can sense sounds, vibrations, accelerometers (tilt device), temperature, moisture, humidity, motion sensing, distance to objects, IR, RF, RFID, touch of buttons, spoken words, gestures, infrared camera, GPS location, light, dark, radio strength RSSI, eye tracking. And of recent seeing and precisely decoding 2d optical vision markers (*e.g*., barcodes and QR codes) — more on that later using my Semantic Marker™ and the **PAT** (Point at Things) section below.

IoT controllers are local edge nodes that interact with the sensors (internal or physically connected and reading the sensed values.) These processors can make decisions (too hot, too close, to wet, too dark, just right) for when to wrap relevant values to send via messages to interested parties (using JSON over MQTT), and how to display values (on potentially resource constrained screens), use sounds (words or beeps), vibrating haptics or Augmented Reality (AR) Graphic overlays.

More powerful devices, such as smart phones, tablets and watches, can also contribute as controllers and sensors, while enjoying more screen real estate, better camera sensing and augmented reality (AR) graphic feedback (described below.) This functionality is captured in *apps*. These too can message over Bluetooth and MQTT, while adding HTTP (web), email, texting, FaceTime, Vision Goggles, etc.



IOT CONTROLLER (M5) WITH MOTION SENSOR AND TOUCH BUTTON PHYSICALLY CONNECTED WITH GROVE WIRES. DISPLAY HAS LIMITED SPACE. 3D BUTTON WRAPS RED BUTTON FOR EASIER TOUCH.

Message Framework (BLE, WiFi, MQTT, HTTP, RPC, web-sockets, etc)

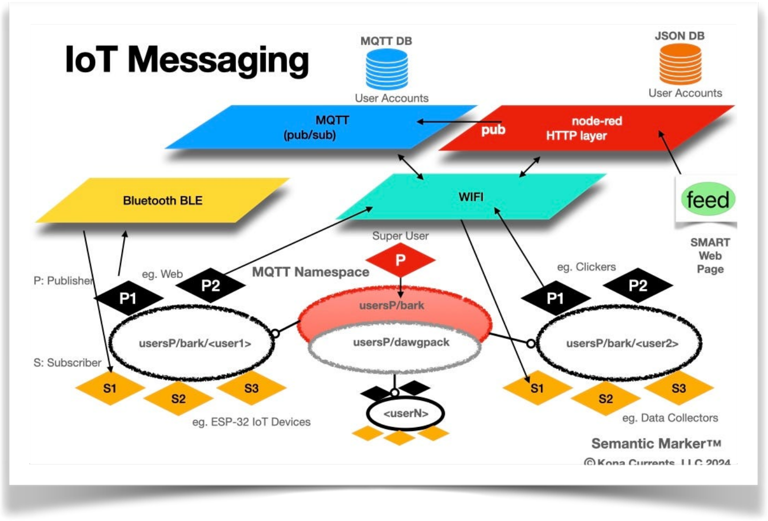
Messages are everything! Without messaging IoT things are almost useless or at least isolated. This communication, hopefully in real-time is accomplished with some kind of transmission medium (radio wave, CB, Wireless, WiFi, BLE, Bluetooth, Smoke Signals, ethernet wires, Optical Markers, Gestures, etc.) Aside from smoke signals, as early as 1799 an Optical Telegraph was created for Napoleon. This could send (relay) information over long distances, but still a coding language was needed. Morris Code was 30 years later and languages that can be accurately processed continue to evolve.

Today, with a renewed data centric focus, we code data values using a language format like JSON. They are sent over a communication medium, we collectively call *messaging*. We send a message to an object, and that message is written in a known language.

> **The essence of what sets Computer Science** apart is the ability to concisely processes messages (or code) controlling the "utter generality of bits"—as Google's Vint Cerf says. In particular the compiler is the main Computer Science tool that translates language (messages or code) into executable bits, over and over without ambiguity.

I could go on with the compiler discussion, believe me, but for this writeup, the compiler is the tool critical for parsing and processing the language constructs, and implementing the concepts sent in the messages. This is the power of *Computer Science*: the efficient translation of a structured language into an executable program.

Application of language translation leads to efficient IoT messaging.



IOT MESSAGING, MQTT NAMESPACE AND USER STORAGE LOCATIONS (IN THE CLOUD)

The diagram shows how elements of the messaging connect from edge devices to the cloud services. Messaging is encapsulated in the following: Language, Transport, Security, and Application Hooks.

- Message **Language** Design (JSON, BNF, IDL, SQL, #hashtags)
- Message **Transport** Capabilities (MQTT, Bluetooth, WebSocket, RPC, DDS, CORBA, Telegram, Discourse, Twitter, Smoke Signals, Telegraph, email, HTTP, WiFi)
- Message **Security** (transport encryption, passwords, tokens, user accounts, namespaces, SSL, location)
- **Application Hooks** for support of messages (e.g., publish & subscribe, REST and node.js)

The resource constrained IoT devices at the edge are mainly using MQTT and Bluetooth BLE many with a JSON format. Every time they start, they try reconnecting with know credentials. When working, all the devices are connected and collaborating with the rest of the collective.

Bluetooth BLE

The power of Bluetooth Low Energy (BLE) is that the IoT devices can act as both senders and receivers of small messages. While they can have many connections, the management complexity has our version staying one for one. BLE doesn't need WiFi and in well defined environments, devices can automatically connect with the first device they find. Assuming only one device (like a IoT Dog Feeder) and one controller (the red M5 smart clicker shown above) — they can connect and talk to each other instantly (and stay connected while in range.) — note that security is complicated and usually non-existent (much like your spouse connecting over your bluetooth speaker while you are using it.)

MQTT pub/sub require a constant runtime presence (and WiFi)

Subscribing for information on a password protected topic requires a constant running presence (basically so messages can be received.) This requires a runtime library which general web pages don't support. But almost all apps and the IoT edge devices support the MQTT protocol. As mentioned above, the ability to talk to any edge node in the world requires them to be listening, subscribing, and a message language designed to find a route to every edge node (and why every device needs an addressable name discussed below.) User protected topics are how our information is kept separate unless group topics are shared for party-line chatting.

But first every edge device needs a WiFi network as MQTT is basically a tunnel through WiFi — and timing is important (as WiFi needs to be connected before MQTT can be connected.)

Login Credentials

To get Wifi running is more complicated, as it needs the *credentials* of the network. These are the SSID and passwords you enter at a new facility (or your device remembers.) On top of that, the MQTT publish subscribe system also require login credentials. Thus BLE is a good way to provide these credentials assuming you have an app that can pass upwards of 100's of characters.

The other method is using what's called an A*ccess Point* (AP) — where the ESP-32 IoT device turns into a web server and hosts its own WiFi network. You use that WiFi and the AP brings up a web page — served by the IoT device. When done, the device stores those values and reboots.

Credentials using the JSON approach might look as follows:

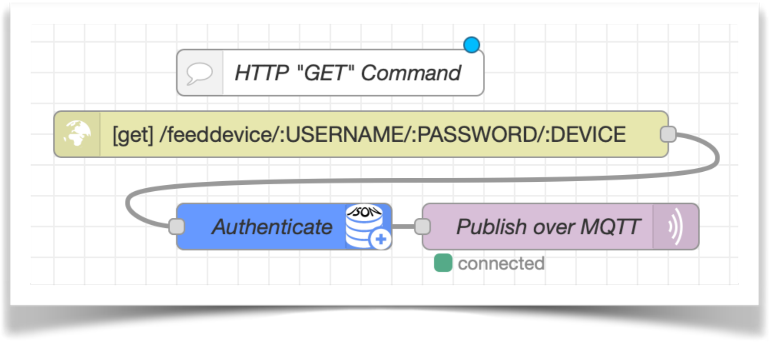
```
{"SSID":"myWifi",
"SSIDpass": "wifiPassword",
"username": "MQTT username",
"password": "MQTT Password",
"device": "DeviceName"}
```

Processing would first connect to the WiFi, and then connect to the MQTT server and start receiving and sending messages.

Web Calls can run almost everywhere

The power of the HTTP web protocol is that almost any browser or device can call a web page. These GET and POST protocols send information to a known address represented by a server processing the HTTP messages.

Node-red is a valuable web server approach acting as endpoints for web calls. Since this constantly is running, it stores user state, and it can also communicate with other processes — in particular the MQTT messaging substrate. Thus a web call for *feed*, as shown below, would have the username and password added as parameters (on the URL.) The node-red processing (shown graphically) would take those parameters and send the *feed* message over the MQTT network addressed to the device specified.



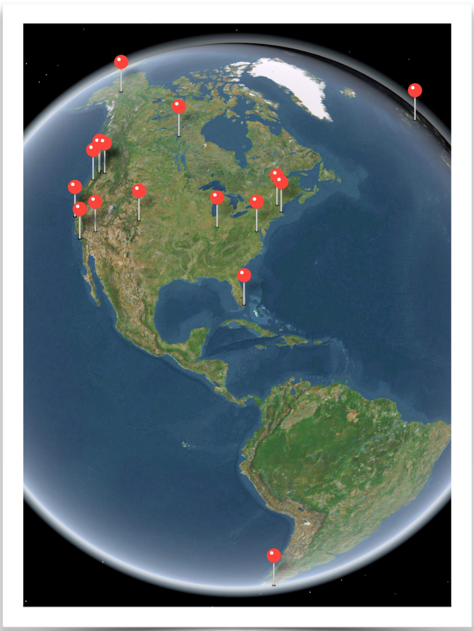
node-red endpoint for a "feed" command passing the username, password and device, authenticating, then publishing over MQTT

This is securely called via a URL as follows (to a fictional my-node-red.com server.)

> https://my-node-red.com:1880/feeddevice/myUser/myPassword/myDevice

Internally to node-red, this is authorized and then sent through the MQTT messaging framework (we are using mosquitto.) The round trip from the web call, to the cloud, and then back down to the devices at the world's edge is within a second (assuming all have fast networks.) *Impressive*!

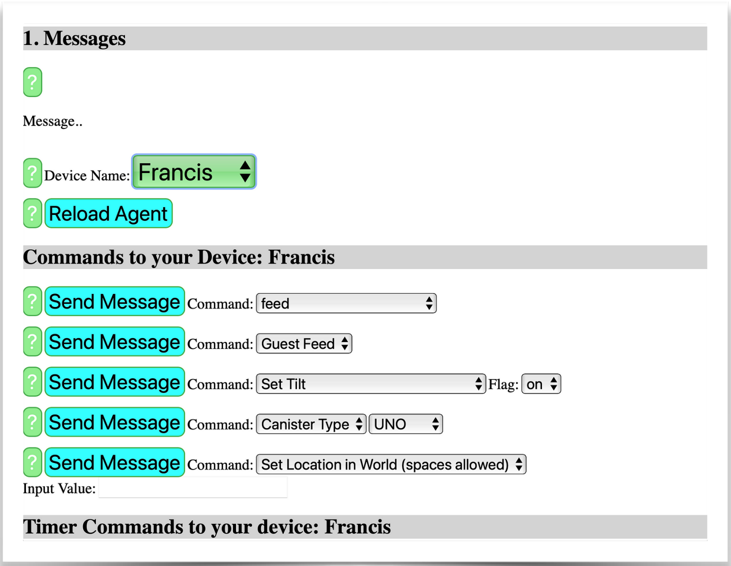
The map below is a real-time update after a message was sent to all our IoT devices running world wide. Their status and general location (city) are returned and displayed — in real-time (if the user opt'd in.)



THE REAL-TIME MAP SHOWS SOME OF THE IOT FEEDERS DEPLOYED WORLD WIDE (REAL OR SIMULATED BASED ON OPTIONALLY USER SUPPLIED CITY NAMES.)

Web Pages wrap all the messaging functionality

Once messages have been designed and coded into all the IoT devices, wrapping in a custom made and authenticated web page is powerful. A web page, as partially shown below, can be run on any device even a watch. Subsets of these messages can be extracted into smaller pages as well (maybe just the *feed* command.)


1. Messages

?

Message..

? Device Name: Francis

? Reload Agent

Commands to your Device: Francis

? Send Message Command: feed

? Send Message Command: Guest Feed

? Send Message Command: Set Tilt Flag: on

? Send Message Command: Canister Type UNO

? Send Message Command: Set Location in World (spaces allowed)

Input Value:

Timer Commands to your device: Francis

SMART WEB PAGE TO SEND MESSAGES (SUCH AS FEED) TO THE DEVICE (HERE NAMED FRANCIS). SEE MAP.

It's amazing how must *javascript*, *html* and *css* it takes to make this powerful message passing capability.

Names for everything

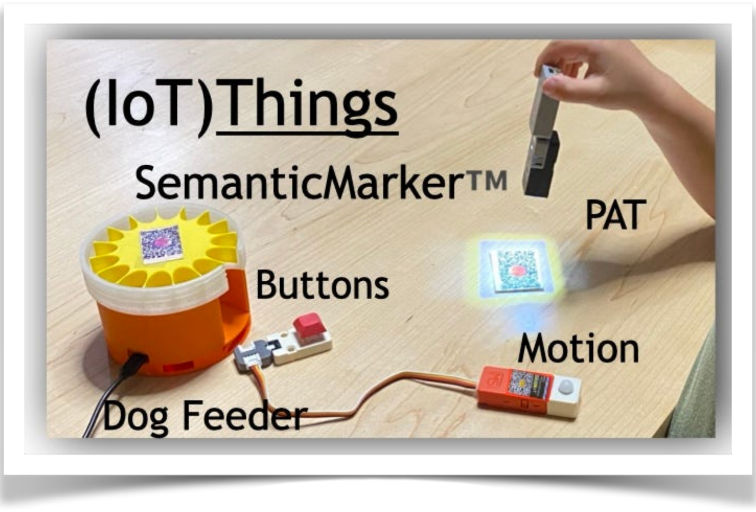
Sending a message to a particular object requires a way to specify that object. Here naming is invaluable; everything needs a distinguishing name. All the IoT devices shown in these images have a name. Sending messages through MQTT can either use fine grained topics, or do what many IP multi-cast approaches use: encode the naming in the messages.

Of particular importance is when multiple devices are running; there needs to be ways to distinguish among them. *"Siri, turn on the living room light"* is complicated when there are lots of lights and switches.

WHICH LIGHT IS WHICH? NAMES WOULD HELP. "SIRI, TURN ON LIGHT #3, COUNTING FROM LEFT, OR FROM RIGHT IN SOME COUNTRIES LANGUAGES"

PAT -Point at Things (Optical Vision Markers)

With the *messaging* foundation described, the ability to trigger devices through a URL, and IoT devices listening for "trigger" commands — offers a new paradigm of invocation — one particularly useful for the #Abilities market. Through our (various) international trademarked **Semantic Marker™** concepts, special optical visual markers, we call a S**emantic Marker™** such as t*wo-dimensional optical code images* and barcodes, *come alive* through interactive feedback with the environment; continual scanning will send secure internet messages to IoT devices and even show unique interactive *Augmented Reality* graphic overlays based on contextually and semantically relevant information (can you say *Minority Report*!)



PAT—POINT AT THINGS (SCANNING SEMANTIC MARKER™ TO TRIGGER IOT THINGS THROUGH MESSAGING)

One can just **Point at Things** — or **PAT,** and the correct IoT feeder is triggered, or the right light turned on. Using a **Semantic Marker™** aware scanner (like the ones above and below) or a smart phone app, optical visual markers can be scanned and messages sent to the devices encoded in the image.

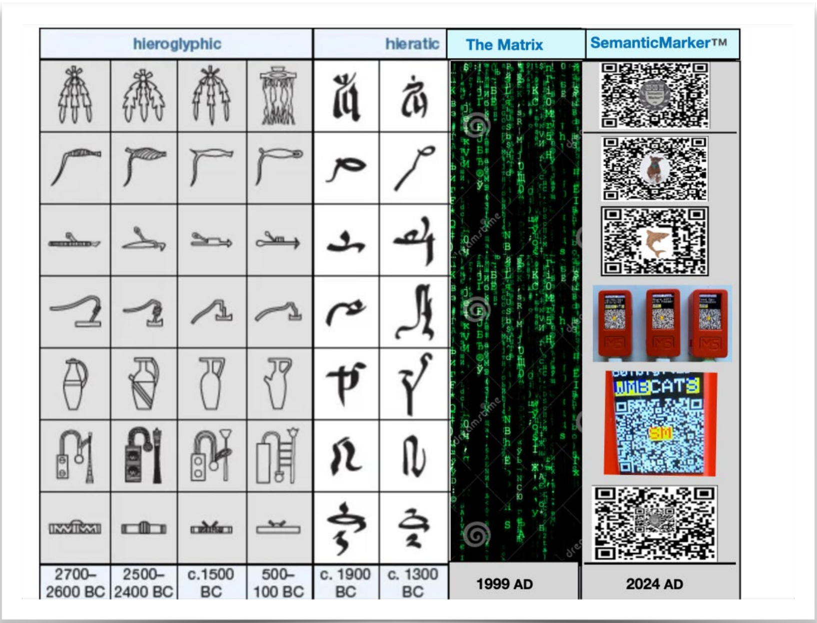
PAT SCANNER FROM SEMANTICMARKER.ORG—3D PRINTED CUSTOM CASE FOR IOT PARTS (AUTHORS PIC)

Aside from triggering IoT devices, scanning optical markers can also be part of renewed *agency* by linking together or grouping common tasks. We call this a *Vision Linker* where operations, described with a **Semantic Marker™**, are collected (linked) for future use. So scan the actions (like feed the dog, turn the tv channel, or multi-step actions) — and store them away for use in the future. #Abilities programming

More on this in a follow on article (and please visit SemanticMarker.org).

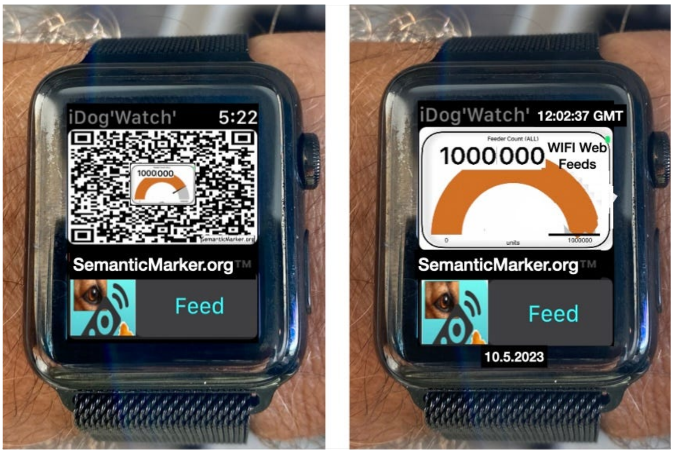
Augmented Reality (AR)

The **Semantic Marker™** is the anchor for creating a new user interface interaction. With just two example below, the smart watch shows the 2d optical vision marker, and then switches to the graphic image behind the marker. This is done both with an



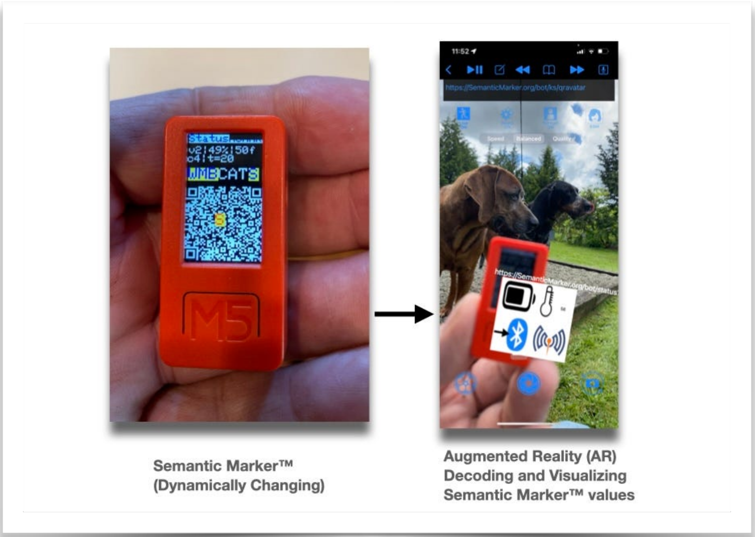
THE 2D OPTICAL VISION MARKER—THE SEMANTIC MARKER™— ARE THE NEW HIEROGLYPHICS (BUT ONES WE KNOW HOW TO DECODE AND LEVERAGE THROUGH MESSAGING)

Augmented Reality (AR) graphic overlay, or just substituting the image with same AR graphic rendition (either static or dynamic.) We call these the *Avatar* behind the **Semantic Marker™** .



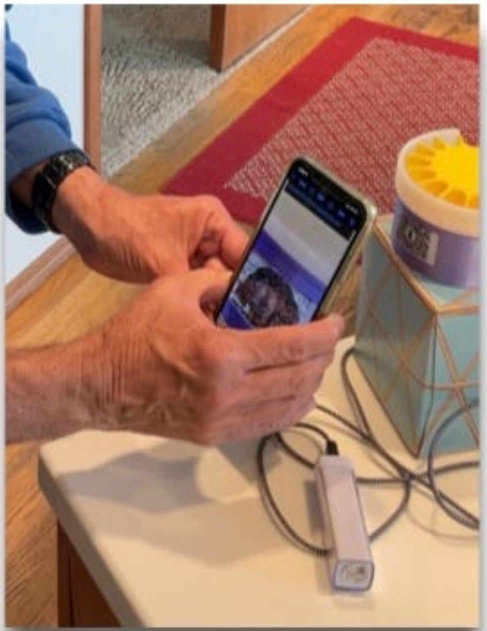
AUTHORS APP FOR THE SMART WATCH SHOWING AUGMENTED REALITY (AR) IMAGE OF NUMBER OF GLOBAL FEEDINGS OVER WIFI TO THE ESP-32 DOG FEEDING IOT DEVICES (1 MILLION AND COUNTING). CLICKING FEED BUTTON TRIGGERS IOT FEED

The IoT devices like the M5 shown below, can also create and show a **Semantic Marker™**. These can be dynamically updated and with the right decoder (scanner) the meaning and values behind the optical marker can be displayed (and constantly updated.)



(LEFT) SEMANTIC MARKER™ DISPLAYED ON M5 IOT DEVICE, CONSTANTLY CHANGING. (RIGHT) USING MY IOS APP, THIS IS DECODED AND DISPLAYED USING AUGMENTED REALITY (AR) GRAPHIC OVERLAYS. A NON APP AWARE TOOL WOULD JUST TAKE YOU TO A WEB PAGE (TRY IT)

With this hands-off displaying of valuable information, a unique new form of user interface is possible, one especially suited to the #Abilities market. The image below shows this where a "trigger" message to the dog feeder just by scanning (not touching) the optical marker — again, regaining **agency**.

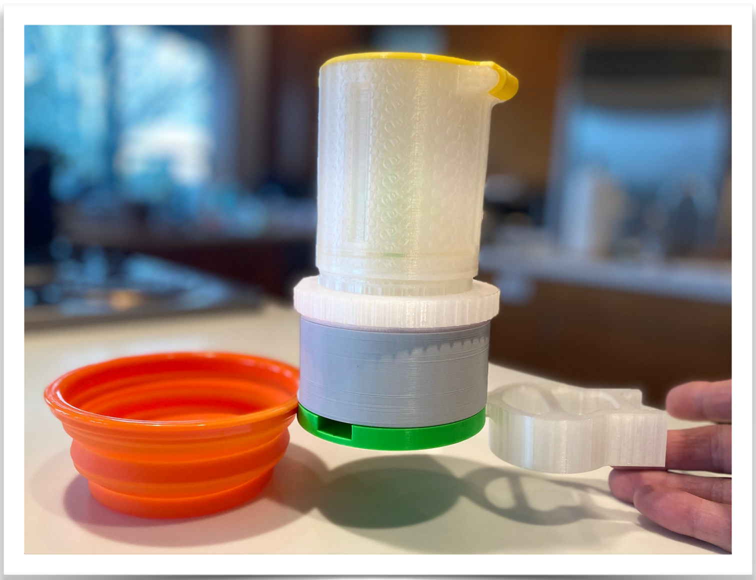


#Ability: Semantic Marker™ Triggers Feeder automatically without clicking (This app shows Augmented Reality overlay of dog)

A non technical user doesn't even have to touch the iPhone, with **PAT** it automatically triggers the dog feeder.

3D Printing and not afraid to use it

Finally pulling all these together and deploying will help the **#Abilities** users. With the right skills (thanks Orion), designing 3d models and printing with 3d printers is a powerful capability. The sky's the limit, from unique mounts for the various wheelchairs, new easier to touch buttons, or unique containers for any of the IoT devices.



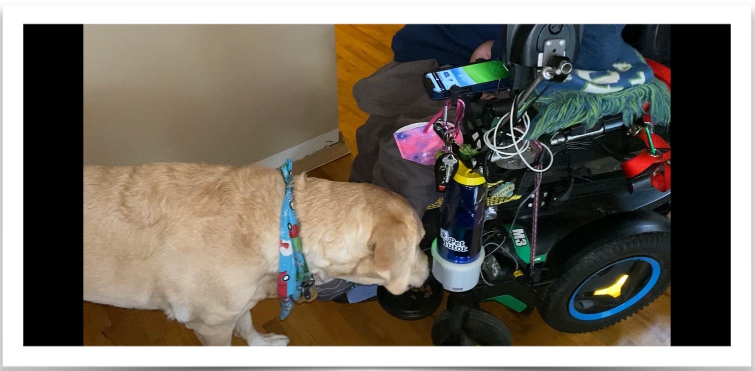
3D PRINTED DOG FEEDER WITH CUSTOM WHEELCHAIR MOUNT (ON RIGHT) AND EMBEDED ESP-32 CONTROLLER (LISTENING OVER BLE AND MQTT). (PRODUCT CALLED PETTUTOR FROM SMART ANIMAL TRAINING).

Putting it all together for #Abilities

The elements discussed offer a powerful capability for anyone, but can especially help bring back *agency* to the #Abilities market.

The IoT devices are almost tailor made for hands-off use. For example the motion sensor lets a user wave their hand to *trigger* a device (feed the dog, turn on the lights, turn the wheelchair, etc). A proximity sensor can let the user know about close objects (maybe sounding off or vibrating.) The potential is unlimited.

When combined with the IoT messaging framework outlined above, new imagined interaction is possible between the IoT sensors, IoT controllers, the users, trusted remote agents and others. Wrap in custom made 3d printed containers, and it's an exciting new opportunity.



DOG FEEDER CONNECTED TO A WHEELCHAIR USING A CUSTOM 3D PRINTED MOUNT (LIKE PIC ABOVE). THIS ONE IS CONTROLLED THROUGH VOICE COMMANDS—"SAYING" NUMBERS (1–10) THAT MAP TO THE IPHONE APP BUTTONS ON EACH SCREEN

Start thinking outside the box and dream up possible solutions using the concepts described here. Drop me a note especially if you are interested in trying this framework as a client. More information at my links.

konacurrents - Overview Computer Scientist, IoT Architect, Boeing Associate Technical Fellow (retired), Adjunct Lecturer and Advisor RTC, ACM…github.com