HAPI System v1.0



Copyright Maya Culpa, LLC

Updated: Jan. 3 2016

Version: 1.2

Contents

[Overview 1](#__RefHeading__154_728816526)

[Near-Term Goals (Non-Technical) 1](#__RefHeading__156_728816526)

[Evolve Collaboration 1](#__RefHeading__158_728816526)

[Establish Key Implementations 2](#__RefHeading__160_728816526)

[Foster Key Partnerships 2](#__RefHeading__162_728816526)

[Create and Provide Educational Resources 3](#__RefHeading__164_728816526)

[Establish The Clean Seed Network 3](#__RefHeading__166_728816526)

[Near-Term Goals (Technical) 4](#__RefHeading__168_728816526)

[Automation Core 4](#__RefHeading__170_728816526)

[Site Controller 4](#__RefHeading__172_728816526)

[Operating Model 5](#__RefHeading__174_728816526)

[Resources and Funding 5](#__RefHeading__176_728816526)

# Overview

HAPI is an evolution in community-driven food production technology. Originally focused on hydroponics (hence Hydroponic Automation Platform Initiative), the program has branched out and now incorporates other food production scenarios such as hoop houses and controlled environments. A guiding principal is that HAPI is not a strictly defined system in a technical sense. The HAPI research and development process is on-going, extending in technical scope and definition as required, in order to most effectively resolve challenges and integrate with other emerging technologies and best practices. In other words, just because a given technology or technique is not part of the current platform does not mean it is an excluded possibility.

With regards to other guiding principles: scalability, flexibility, fault-tolerance and security, these should be concisely and precisely defined for every facet of the program. HAPI is also an *accessible* technology platform as it strives to keep capital investment and operational costs as low as possible. There is little value in being technical elitist and simplicity tends to scale well.

Each HAPI implementation will produce food using different forms of lighting, nutrient delivery systems, substrate materials, structural designs and operational practices. We will eventually have the ability to simulate diverse environmental scenarios such as deserts, underwater, cold weather and space. Materials and construction processes will be tailored towards the resources and capabilities of the typical maker space or home workshop.

In the area of urban food production, HAPI will introduce low cost, manageable systems that can be easily scaled for typical constraints (space, expertise, money and energy) and optimally configurable for quantity, nutritional profile and climactic diversity.

# Remote Terminal Unit Features

## HAPI System v1.0 Firmware Specifications

The HAPI System v1.0 firmware requires:

* Built specifically for the Arduino Mega 2560
* Requires a standard Ethernet shield and access to a DHCP server
* 27.5k of program storage space (11% of that available on the Mega 2560)
* 1.2k of RAM (14% of that available on the Mega 2560)

### Pin Settings

The firmware designates a pin map that both allows for user flexibility and ensures proper hardware operation. The map is as follows:

**Pins Function**

0-1 Reserved

2-3 Analog Output

4-5 Reserved

6-7 Digital Output

8-21 Reserved

22-27 Digital Output

28-49 Digital Input

50-53 Reserved

A0-A11 Analog Input

A12-A15 Reserved

### Default State Management

Digital and Analog output values can be pre-configured and stored in the RTUs on-board EEPROM. The configuration data is specified as. The RTU supports functions to set and read these values from EEPROM.

### Boot Process

### Command Language

# Master Controller Features

## HAPI System v1.0 Master Controller Specifications

The Master Controller function is meant to be implemented on a Raspberry Pi 3. However, most modern computers can be configured to fulfill this roll. This documentation assumes the MC function is on a Pi. A HAPI-specific image for the Raspberry Pi is available, eliminating the need for the user to have to install or configure anything at all in order to get a HAPI System v1.0 Master Controller up and running.

Site Validation

The first task the Master Controller performs when it starts is to validate the site. This task consists of two steps: RTU Discovery and Pin Mode Validation.

RTU Discovery

In order to minimize system configuration on the user’s part, the Master Controller can self-discover remote terminal units that are functioning on the same network. (Note: this is accomplished using hard-coded MAC addresses and a program called “arp-scan”.)

During this phase, the Master Controller prints out a list of all RTUs that is discovers.

Pin Mode Validation

After discovering RTUs, the master controller validates the pin modes of each online RTU with a map of expected pin modes for that RTU. Suppose for instance that the database reports that pin 3 on RTU1 should be configured as a DigitalOutput. The Master Controller asks RTU1 for its pin modes. If the RTUs pin 3 is indeed configured as a DigitalOutput, it’s a valid match. However, if the RTU reports a pin mode that differs from what the database says, it’s reported as an issue.

Pin mode validation is important because if the pin modes on the RTU do not match what is expected, sensor readings and control functions cannot be relied upon.

The Scheduler

The Listener

Logging

# Analytics and Visualization

## HAPI System v1.0 Analytics and Visualization

The Master Controller function is meant to be implemented on a Raspberry Pi 3. However, most modern computers can be configured to fulfill this roll. This documentation assumes the MC function is on a Pi.

Real-operations Dashboard

Historical Trend Visualization

Visualizing the Command Log

Tabular Data