

# From Turnkey To Custom

---

Hacking on a MoreBeer Brew Sculpture

nate@yocom.org

Who am I?

---

# The Journey to a Brew System

To Turnkey, or to Build...



VS



# Turnkey!

## 1 - Company

Quite simply, MoreBeer rocks. Fantastic folks, with homebrew spirit, making awesome products here in the USA. I.e. it's still handmade quality shit!

## 2 - Requirements

- Stainless
- Oversized kettles
- Pumps
- Easy to use, and to clean
- Balance of automation, and manual control

## 3 - Price

Not for everyone, but compared to my neighbors "classic" car fetish ...

# The System

Before the hacking ...



# Out of the Box



# And into the brewery...



- Oversized kettles (10-15 gal brews)
- Convoluted Coil HERMS
- Sparge Arm
- High Temp Silicon Tubing
- MT False Bottom
- March Pumps
- Chiller
- Boil Whirlpool
- ...

# Last but not least ... The Panel





Commence The Hacking!

---

**DANGER!**

**ACHTUNG!**



---

# Physical Hacks

---

# Brew Day! Setup Step #1

**Pull System Out**

**Roll the system from interior of garage to outside door...**

**Realize Garage is NOT Level**

**Drainage angle good for rain, bad for mash/boil!**

**Needs to be leveled!**

**Where'd I leave that level? With or without pots?**

Enter the RV Level!



# Double Brew Days == Burner Madness

Brew #1 gets to Boil

As first brew hits boil kettle, next wants to be mashing in....

Brew #2 Mash In

MT Flame @ 100% and  
HLT @ 100% + Boil start  
...

So little flame control.  
Why? Ball valve!

SO MUCH FLAME!  
SO LITTLE CONTROL!



# Enter Pin/Needle Valve!



- 1/2" NPT M/F conversions on either side of a 1/4" NPT needle valve
- Finer control for **just** boil burner
- Adjustment to burner air intake was necessary to tune flame

# Software Hacks: The Panel

---



# What is inside?



- Arduino Due:

<https://www.arduino.cc/en/Main/arduinoBoardDue>

- Custom Daughter Board
- DIABLO16 based display:

[http://www.4dsystems.com.au/product/uLCD\\_70DT/](http://www.4dsystems.com.au/product/uLCD_70DT/)

# How does it work?

- Arduino contains all logic and controls all flow and measurements via:
  - Serial connection to screen
  - A/C relay's on daughter board for Pump1, Pump2
  - GPIO for HLT burner (high/low signal)
  - Dallas OneWire digital temperature sensors for HLT, Mash and Boil Kettle
- Screen handles all touch interface display and feedback with small app running on display processor.

# I Can Reprogram This Stuff!

More of the “smarts”

Wouldn't it be cool if I could extend and automate my brew day, record everything, play it back, raise my mash from the back porch or get alerts on my phone as things happen?

Maintain the manual

However, there are days I don't want all the crazy automagic stuff, KISS and just make beer.

Needs Wireless!

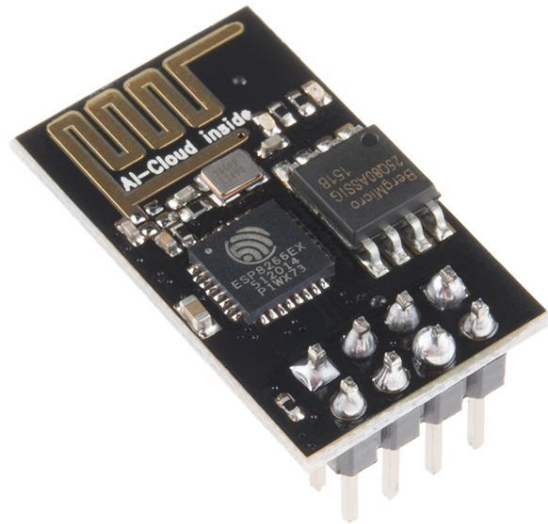
Why not add wireless to the system, and “move” the smarts from the panel's internals, to a client application which could run anywhere? Leave the interaction on the panel to the basics necessary for a manual brew day.

# Step #1 - New Software - The Plan

- Create a simple on-board/out-of-box UI for **basic** automation of a brew day
- Add wireless capability such that the sculpture can:
  - Operate as it's on AP and/or join an existing wireless network
  - Be monitored or controlled by an application running on a separate device
- Create an open source library for interacting with the brew sculpture
- Create a desktop or mobile application for automation including:
  - Recording a brew day and all measurements for later 'replay' and analysis
  - System calibration and testing
  - Automate/replay brew days from recipe imports

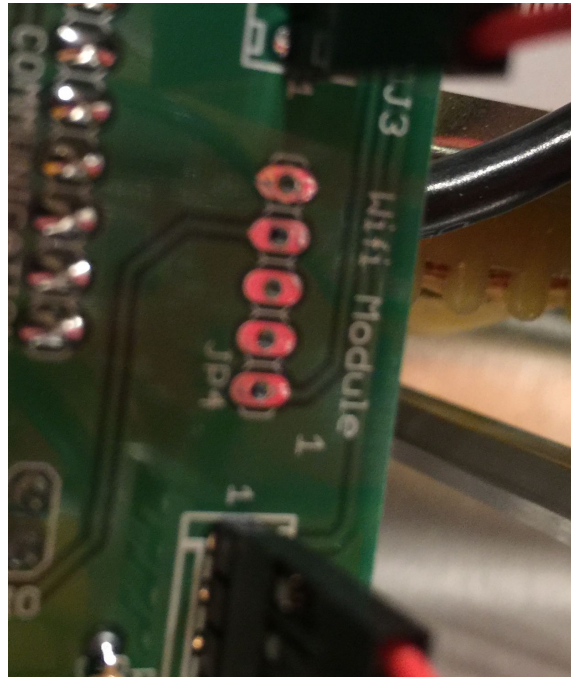
## Step #2 - Needs Wireless!

- Find and purchase a simple wifi module compatible with board:
  - ESP8266 => <https://www.sparkfun.com/products/13678>
- Simple serial interface and 3.3v requirement make it ideal for Arduino Due

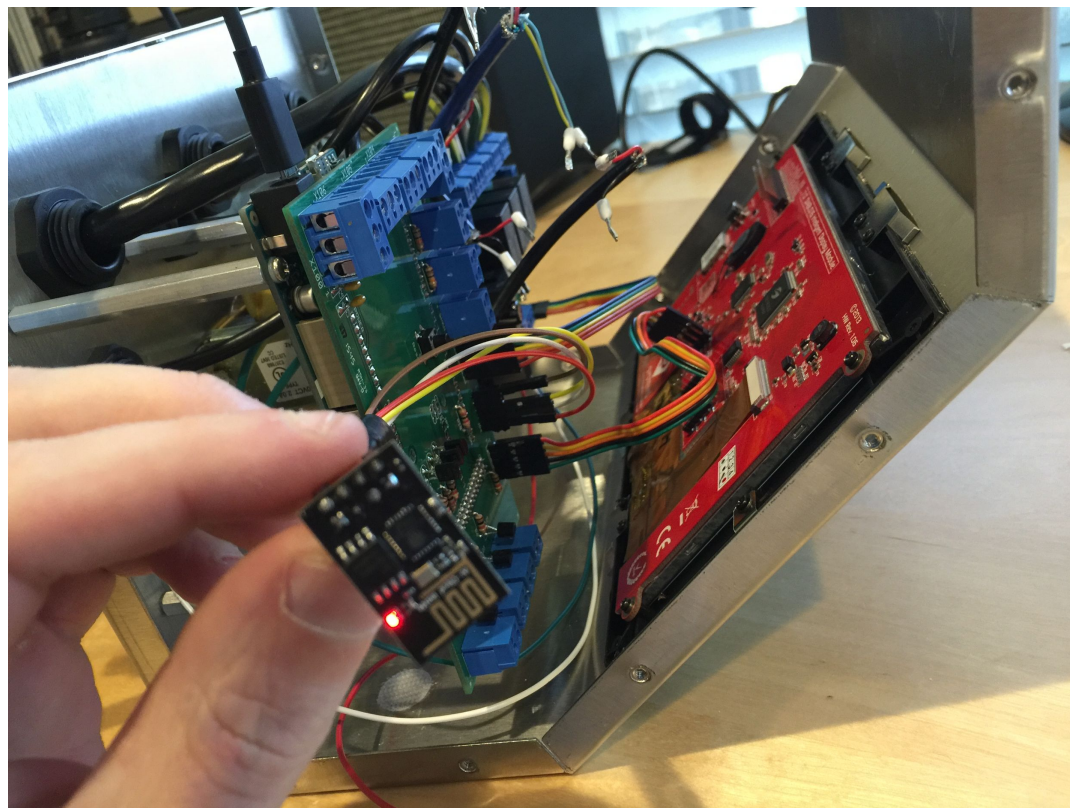


## Step #2a - Needs Wireless - Connect It

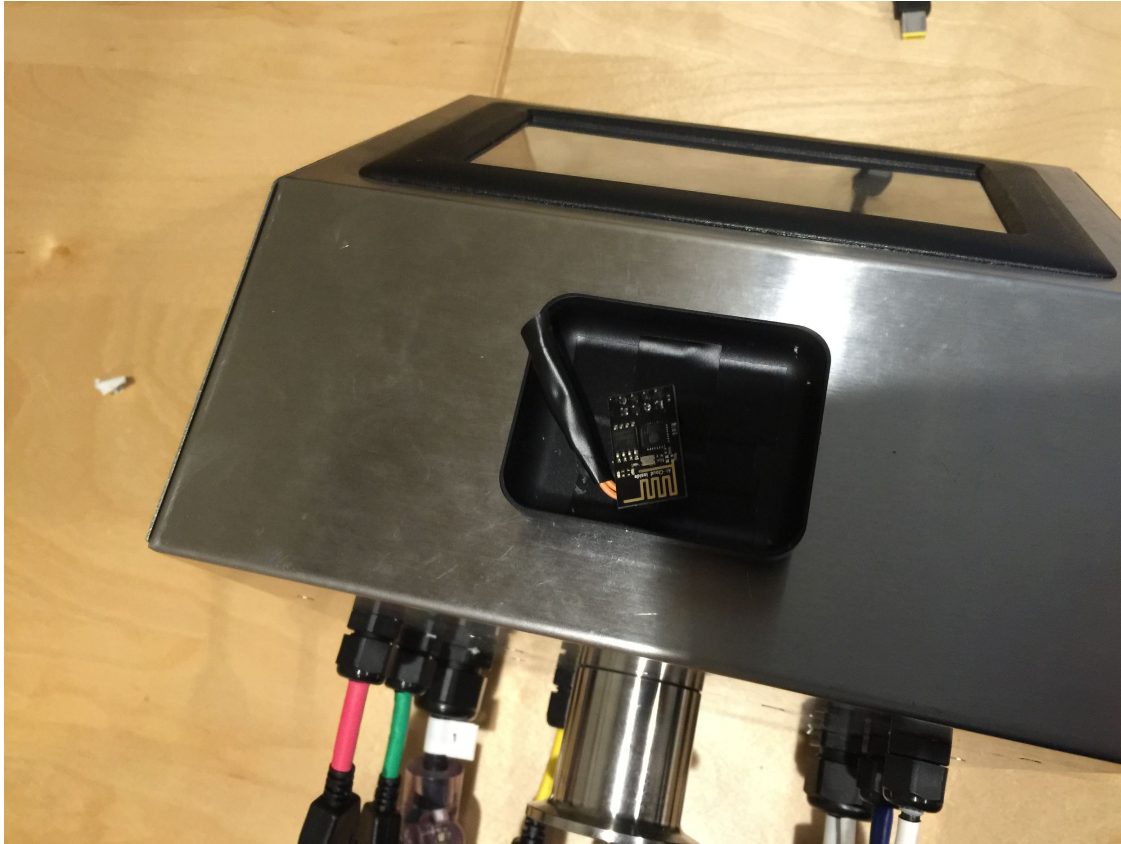
- MoreBeer thought of this, and conveniently included solder points on their daughter board which pass through to the arduino's serial and power interface:



We Have Power!



# Stainless Enclosure != Good Wireless



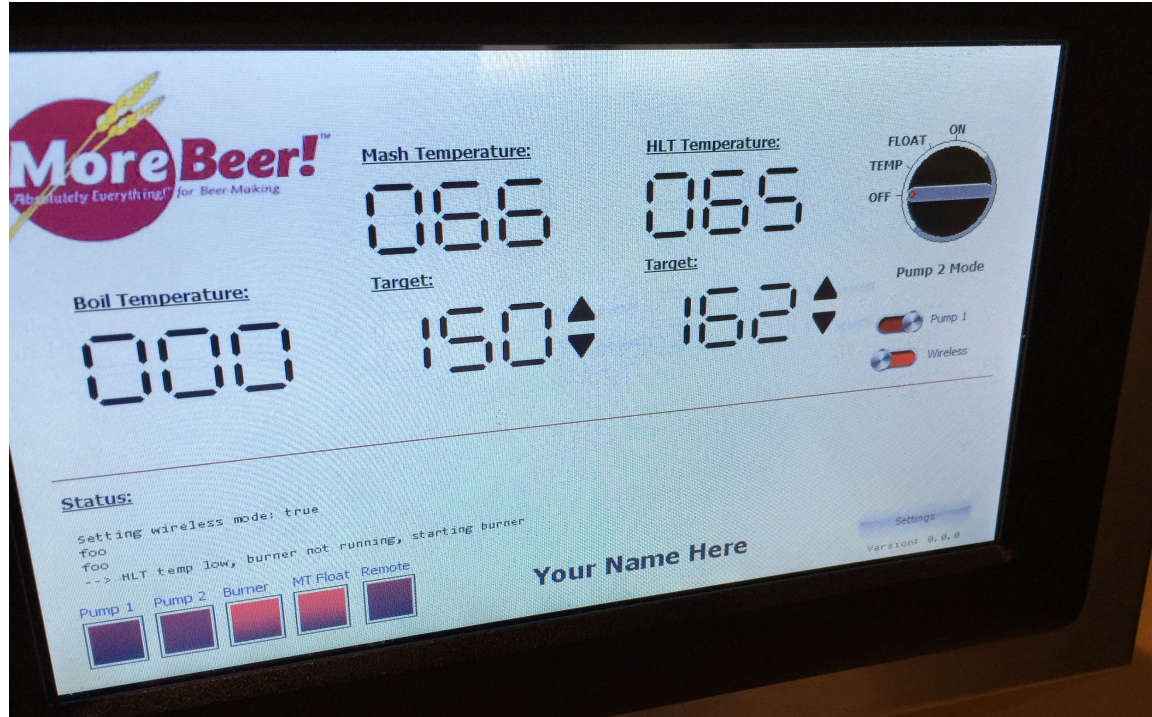
- So drill a hole!
- Clamshell headphone case as enclosure



# Other Challenges

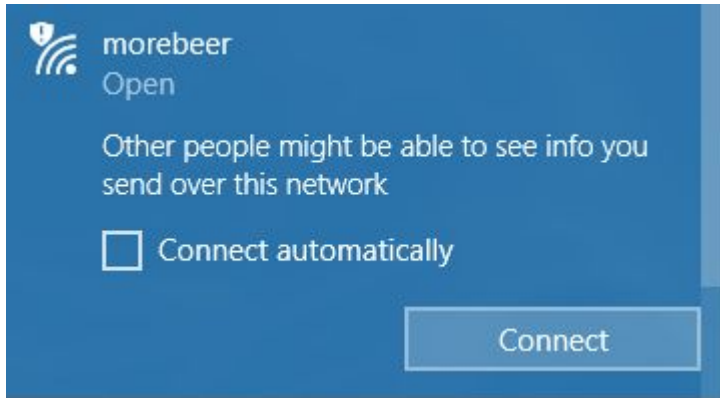
- Arduino is slow (~80Mhz) - serial buffer overrun
  - Eventually set ESP8266 to 9600 Bps
- Idempotency
  - Eventually settled on small broadcast of status with high degree of repetition to cover missed messages, as opposed to send/ack control
- Security
  - Default mode is wifi off, expectation is that user will enable, attach and configure for their AP. Once on specific AP, no additional security (being on same wifi == good enough)
- uLCD-70DT Sucks. Great looking, terrible programming experience.

# The “Mantis” Interface - Control Panel



- Always on Temp indicators
- Target control
- Pump 2 Automation Select
- Pump 1 on/off
- Wireless on/off
- Status Lights and Text

# Getting Online



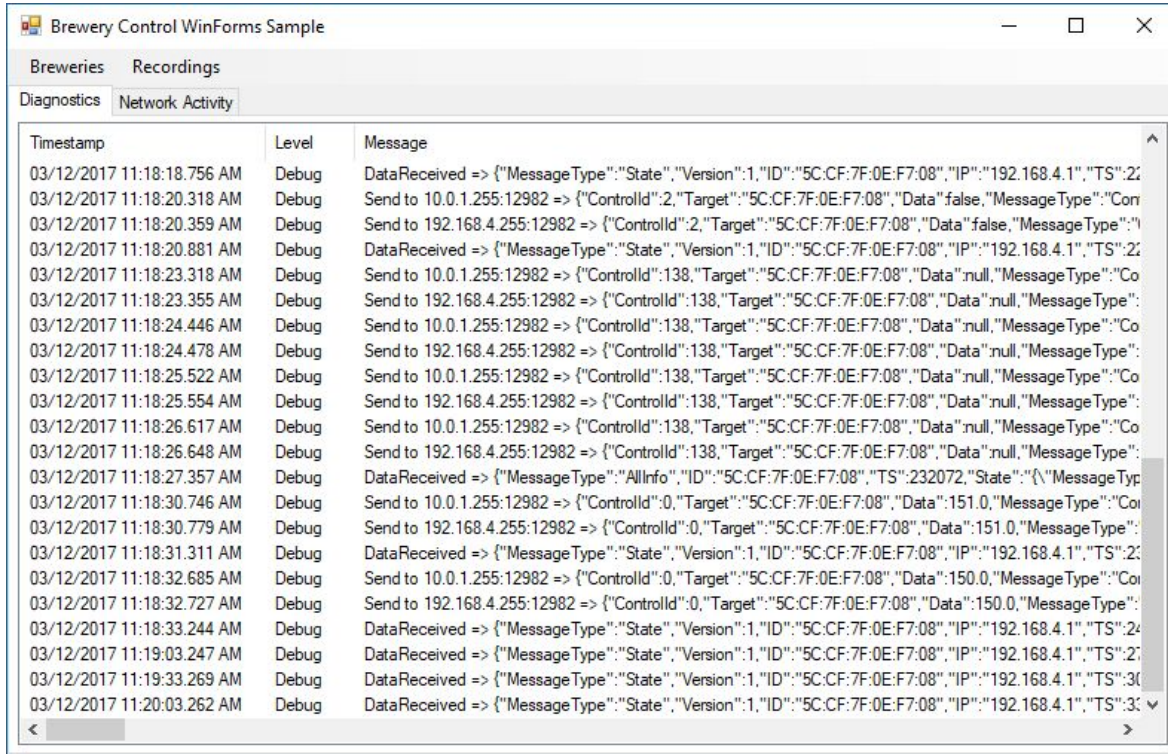
- When wireless is on:
  - Defaults to AP mode as “morebeer”
- Accepts control messages to set wireless network from application

# The Mantis Software - Current State

- ✓ ● Create a simple on-board/out-of-box UI for **basic** automation of a brew day -
- ✓ ● Add wireless capability such that the sculpture can:
  - ✓ ○ Operate as it's on AP and/or join an existing wireless network
  - ✓ ○ Be monitored or controlled by an application running on a separate device
- ✓ ● Create an open source library for interacting with the brew sculpture
  - Create a desktop or mobile application for automation including:
    - Recording a brew day and all measurements for later 'replay' and analysis
    - System calibration and testing
    - Automate/replay brew days from recipe imports



# Early Mantis PoC PC Interface - Discovery



The screenshot shows a Windows application window titled "Brewery Control WinForms Sample". The interface has a menu bar with "Breweries" and "Recordings". Below that, there are tabs for "Diagnostics" and "Network Activity". The main area displays a log of network activity with three columns: "Timestamp", "Level", and "Message".

Timestamp	Level	Message
03/12/2017 11:18:18.756 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":2}
03/12/2017 11:18:20.318 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":2,"Target":"5C:CF:7F:0E:F7:08","Data":false,"MessageType":"Co
03/12/2017 11:18:20.359 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":2,"Target":"5C:CF:7F:0E:F7:08","Data":false,"MessageType":"
03/12/2017 11:18:20.881 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":2}
03/12/2017 11:18:23.318 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"Co
03/12/2017 11:18:23.355 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"
03/12/2017 11:18:24.446 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"Co
03/12/2017 11:18:24.478 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"
03/12/2017 11:18:25.522 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"Co
03/12/2017 11:18:25.554 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"
03/12/2017 11:18:26.617 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"Co
03/12/2017 11:18:26.648 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":138,"Target":"5C:CF:7F:0E:F7:08","Data":null,"MessageType":"
03/12/2017 11:18:27.357 AM	Debug	DataReceived => {"MessageType":"AllInfo","ID":"5C:CF:7F:0E:F7:08","TS":232072,"State":{"MessageType":
03/12/2017 11:18:30.746 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":0,"Target":"5C:CF:7F:0E:F7:08","Data":151.0,"MessageType":"Co
03/12/2017 11:18:30.779 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":0,"Target":"5C:CF:7F:0E:F7:08","Data":151.0,"MessageType":"
03/12/2017 11:18:31.311 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":2}
03/12/2017 11:18:32.685 AM	Debug	Send to 10.0.1.255:12982 => {"Controlld":0,"Target":"5C:CF:7F:0E:F7:08","Data":150.0,"MessageType":"Co
03/12/2017 11:18:32.727 AM	Debug	Send to 192.168.4.255:12982 => {"Controlld":0,"Target":"5C:CF:7F:0E:F7:08","Data":150.0,"MessageType":"
03/12/2017 11:18:33.244 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":2}
03/12/2017 11:19:03.247 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":2}
03/12/2017 11:19:33.269 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":3}
03/12/2017 11:20:03.262 AM	Debug	DataReceived => {"MessageType":"State","Version":1,"ID":"5C:CF:7F:0E:F7:08","IP":"192.168.4.1","TS":3}

- Discover “breweries”
- Record and view all activity
- Edit/control a given brewery

# Early Mantis PoC PC Interface - Brewery Control

Brewery Viewer for 5C:CF:7F:0E:F7:08

### Settings

Current Network:

Brewer Name:

Broadcast Rate:

Remote Control

### Visible Networks

Name	Encryption	Signal (dBm)
Yocom Guest	WPA2_PSK	-71
HOME-A472	WPA_WPA2_PSK	-90

### Boil Kettle

Temperature:

### Mash Tun

Temperature:

Target:

Float

### HLT

Temperature:

Target:

Float

Bumer

### Pumps

Pump 2 Mode:

Pump 1

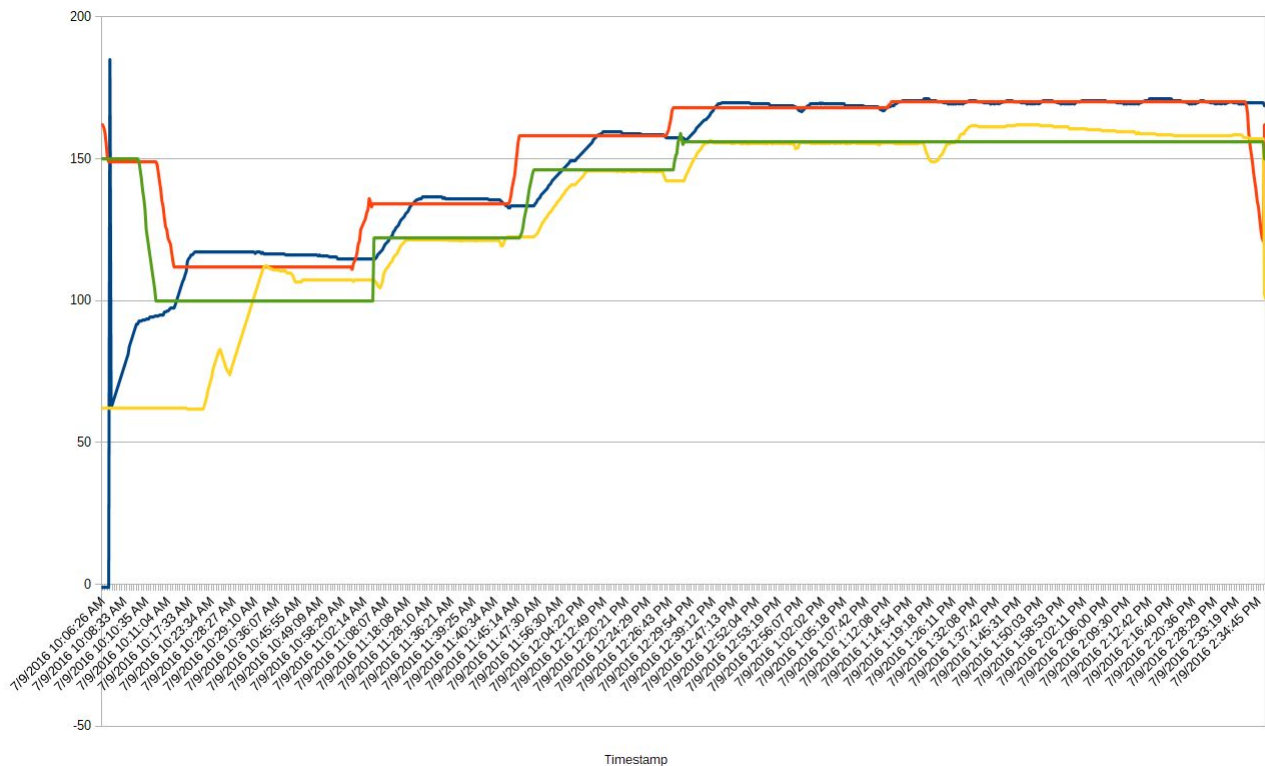
Pump 2

### Status

Timestamp	Message
-----------	---------

- Live view of all settings
- Wireless settings
- Manual automation :)

# Future Smarts!



- Interactive Graphs
- Replay
- Recipe integration and import/replay



# The Status and Future of Mantis

- Available today as open source (BSD) at: <https://github.com/nate-yocom/mantis>
- Personal time/hobby project, happy to share and collaborate
- **Beware flashing your arduino/display** - if you change your mind, you need original source. Chris Graham @ MoreBeer has been very kind during my development - I'd suggest contacting them and offering to pay S&H etc to have your system reset. HOWEVER, this is NOT an implied warranty or service - think carefully before hacking on your own!



# Fini & Resources:

- Me: [nate@yocom.org](mailto:nate@yocom.org)
- Mantis: <https://github.com/nate-yocom/mantis>
- Wifi Chip: ESP8266 - <https://www.sparkfun.com/products/13678>
- Arduino Due - <https://www.arduino.cc/en/Main/arduinoBoardDue>
- uLCD\_70DT Screen: [http://www.4dsystems.com.au/product/uLCD\\_70DT/](http://www.4dsystems.com.au/product/uLCD_70DT/)

- My next project... Fermentation control! ⇒

