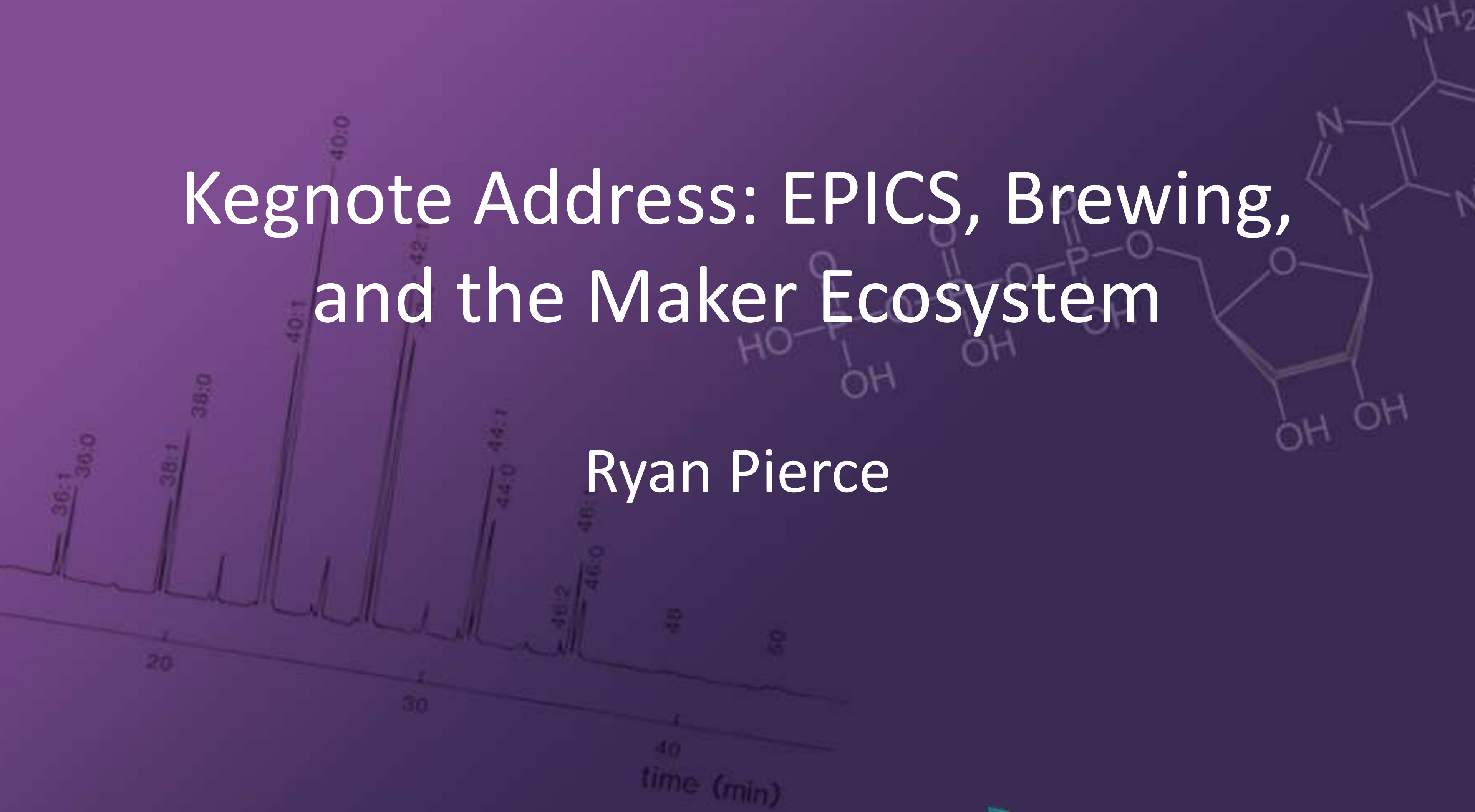


Kegnote Address: EPICS, Brewing, and the Maker Ecosystem

Ryan Pierce

2018 EPICS Collaboration Meeting
6/14/2018

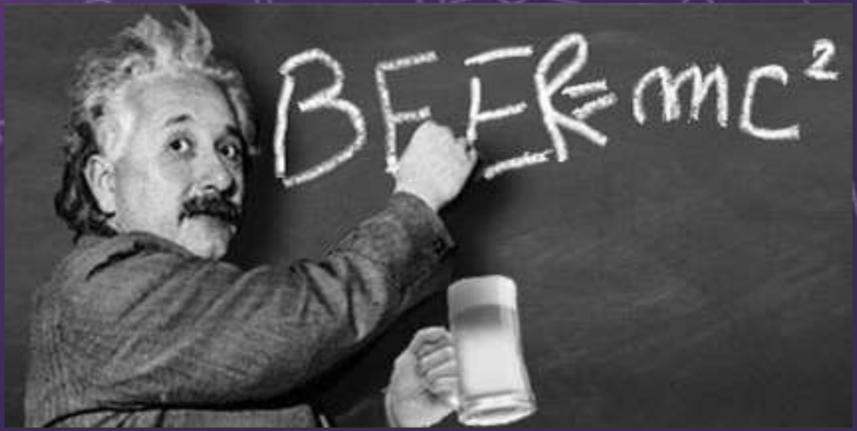
Analytics Lounge

EPICS can control synchrotrons and telescopes. But can it brew beer?

And WHY does someone spend thousands of dollars and hundreds of hours over ~~two~~ four years to build a computer controlled beer brewing system using EPICS?

Because Beer and Physics have a long and distinguished history!

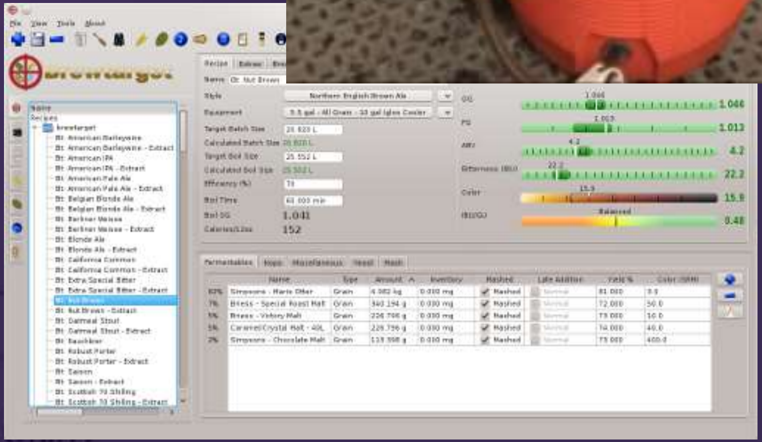
A young Albert Einstein, working for his family's company Elektrotechnische Fabrik J. Einstein & Cie, installed electricity for Munich's oldest Oktoberfest beer tent.



Using EPICS to brew beer carries on this noble tradition!

Because Process Control!

- Brewing depends on careful process control
  - Mash temperature and time control enzymes
  - Goal: Temp. +/- 0.5 F
  - Accurate water volume
  - Boil-off rate
- Process control makes brewing repeatable
  - Beer can be "designed" in software
  - Better recipe development by reducing human error



Name
Recipes
- brewtarget
  - Bt: American Barleywine
  - Bt: American Barleywine - Extract
  - Bt: American IPA
  - Bt: American IPA - Extract
  - Bt: American Pale Ale
  - Bt: American Pale Ale - Extract
  - Bt: Belgian Blonde Ale
  - Bt: Belgian Blonde Ale - Extract
  - Bt: Berliner Weisse
  - Bt: Berliner Weisse - Extract
  - Bt: Blonde Ale
  - Bt: Blonde Ale - Extract
  - Bt: California Common
  - Bt: California Common - Extract
  - Bt: Extra Special Bitter
  - Bt: Extra Special Bitter - Extract
  - Bt: Nut Brown
  - Bt: Nut Brown - Extract
  - Bt: Oatmeal Stout
  - Bt: Oatmeal Stout - Extract
  - Bt: Rauchbier
  - Bt: Robust Porter
  - Bt: Robust Porter - Extract
  - Bt: Saison
  - Bt: Saison - Extract
  - Bt: Scottish 70 Shilling
  - Bt: Scottish 70 Shilling - Extract

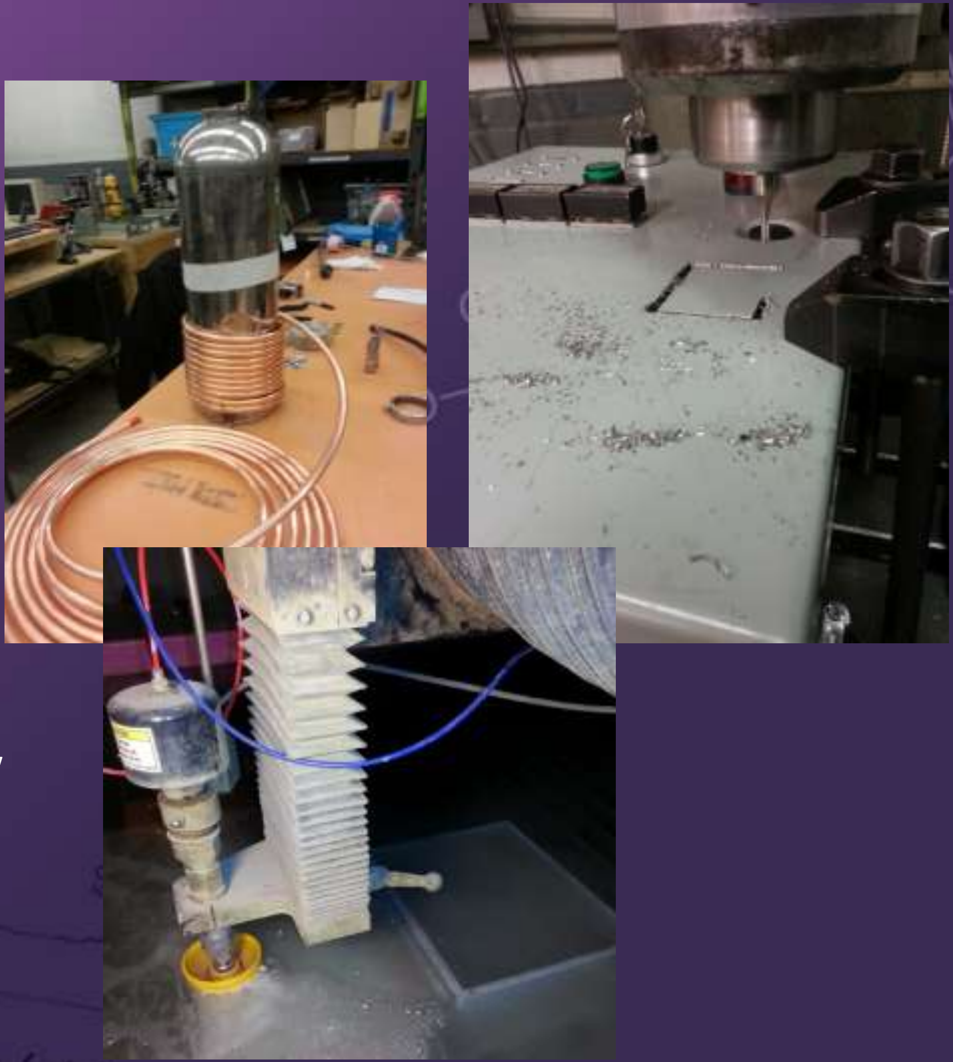
Name: Bt: Nut Brown
Style: Northern English Brown Ale
Equipment: 5.5 gal - All Grain - 10 gal Igloo Cooler
Target Batch Size: 20.820 L
Calculated Batch Size: 20.820 L
Target Boil Size: 25.552 L
Calculated Boil Size: 25.552 L
Efficiency (%): 70
Boil Time: 60.000 min
Boil SG: 1.041
Calories/12oz: 152

OG: 1.046
FG: 1.013
ABV: 4.2
Bitterness (IBU): 22.2
Color: 15.9
IBU/GU: 0.48 Balanced

	Name	Type	Amount	Inventory	Mashed	Late Addition	Yield %	Color (SRM)
82%	Simpsons - Maris Otter	Grain	4.082 kg	0.000 mg	✓ Mashed	Normal	81.000	3.0
7%	Briess - Special Roast Malt	Grain	340.194 g	0.000 mg	✓ Mashed	Normal	72.000	50.0
5%	Briess - Victory Malt	Grain	226.796 g	0.000 mg	✓ Mashed	Normal	73.000	30.0
5%	Caramel/Crystal Malt - 40L	Grain	226.796 g	0.000 mg	✓ Mashed	Normal	74.000	40.0
2%	Simpsons - Chocolate Malt	Grain	113.398 g	0.000 mg	✓ Mashed	Normal	73.000	400.0

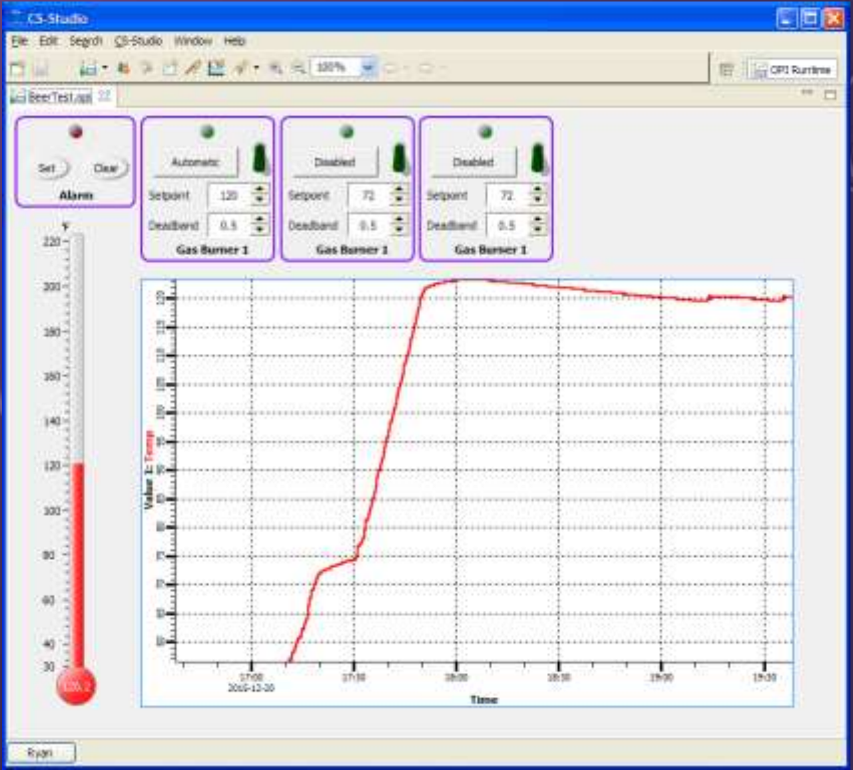
Because Hackerspace!

- Hackerspaces / Makerspaces / Fab Labs provide:
  - Tools (MIG and TIG welders, manual and CNC machining…)
  - Community (people who like to teach what they know)
  - Ecosystem of Open Source, Open Hardware
    - Arduino, Raspberry Pi, BeagleBone
  - Commercial software that is free to makers:
    - Fusion 360 (CAD/CAM), Eagle (PCB design)
  - Encourage the audacious!
- I built this project at:
  - Pumping Station: One (location of brew club, welding, metal fabrication, assembly and testing)
  - University of Chicago Polsky Center Fab Lab (software)
  - Analytics Lounge (waterjet)
  - Lately, a friend's house



Because EPICS!

- EPICS provides much of what I need out of the box
  - Network protocol, code, and libraries
  - Tools to construct operator control panels without writing GUI code
  - Data archiving, web interface
- EPICS lets me focus on my hardware interface and control algorithms



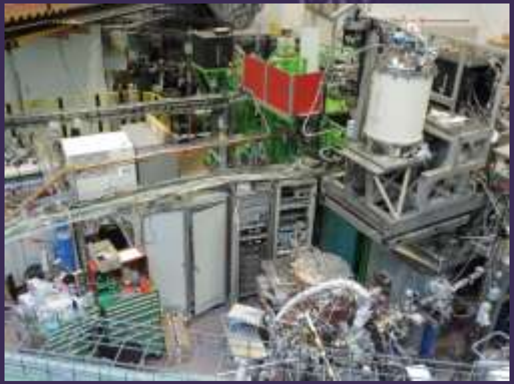
Setpoint 120 Deadband 0.5

Setpoint 72 Deadband 0.5

Setpoint 72 Deadband 0.5

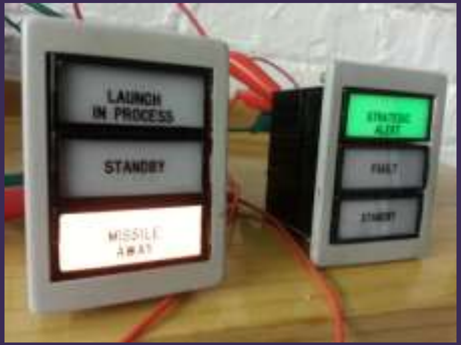
Because Science!

- As a kid, I wanted to be a high energy physicist
  - Lectures and tours at Fermilab, Argonne, working summers at a science museum, intern at CDF
- As an adult, I'm still fascinated by science
  - APS open house in 2012 introduced me to EPICS
  - EPICS community provided significant help and encouragement with this project and enabled access to BESSY II and BER II in 2015
  - Presented at 2016 EPICS Conference at ORNL
- Analytics Lounge – making science accessible
  - Scanning electron microscopy and EDX, gamma and alpha spectroscopy, ICP-MS, pXRF
  - Currently looking for a home for our lab….

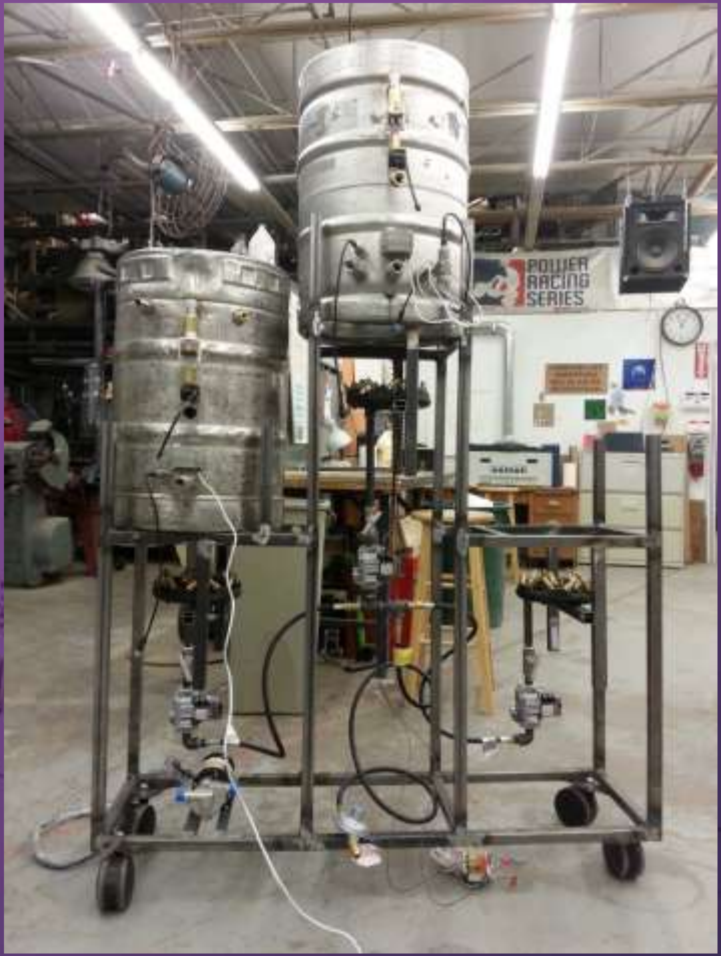


Because Cold War and Nuclear History!

- My favorite vacation spots include Chernobyl, Hanford, US and Ukrainian nuclear missile museums
- I acquired panel indicator lights from two nuclear missile systems:
  - Thor IRBM
  - Minuteman I ICBM
- Swords to Plowshares – repurposing nuclear weapons systems parts for Beer!



System Overview

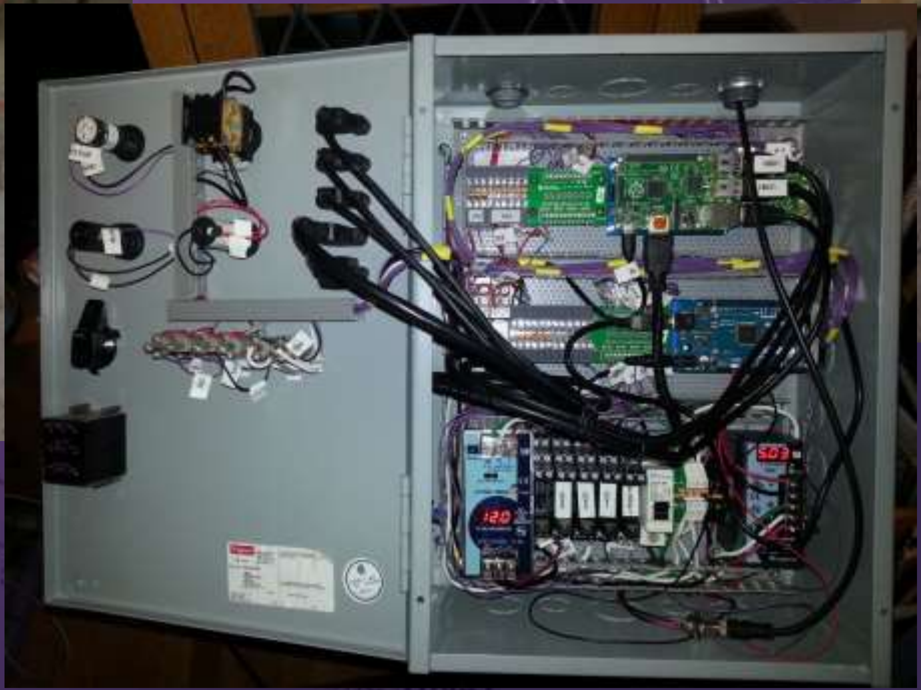


EPICS System Design

- Raspberry Pi running IOC
  - StreamDevice & Asyn via USB serial
- Arduino Mega
  - Interfaces to brewing hardware and sensors
  - Unlike Pi, uses 5 V logic and lots of GPIO and analog pins
  - Unlike Pi, tight timing (bit banging), hardware interrupts don’t require kernel drivers
  - Runs code derived from Pete Jemian’s cmd_response
  - Implements critical control algorithms like firing gas burners without risk of Linux crashing
- CSS BOY operator interface

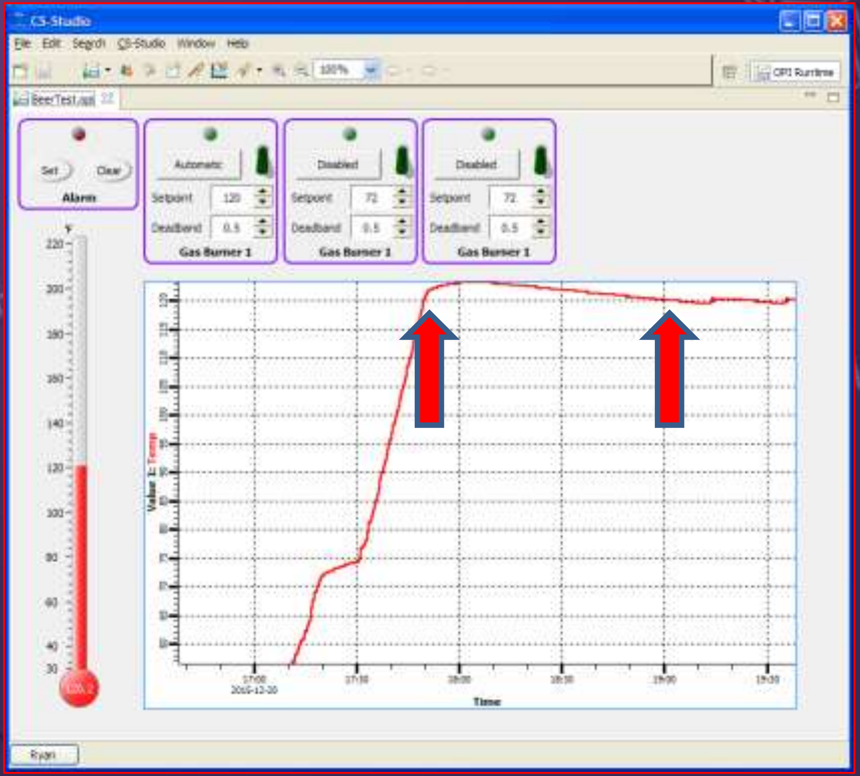
- Actuators:
  - 12 VDC actuated ball valves and solenoid valves for fluids
  - 24 VAC propane burner valves / pilot lights (furnace control)
  - 120 VAC pump
  - Stepper motor (for propane throttle)
- Sensors:
  - Bus of DS18B20 OneWire temperature sensors
  - Analog pressure sensors for measuring fluid levels
  - Flow meters producing streams of pulses

Control Electronics



Correcting Overshoot

- Lag between applying heat via a propane burner and temperature probe responding.
- This lag varies based on amount of water in keg, whether heat exchanger is operating, etc.
- Simple setpoint / deadband doesn't work.
- PID has limited ability to correct for lag.



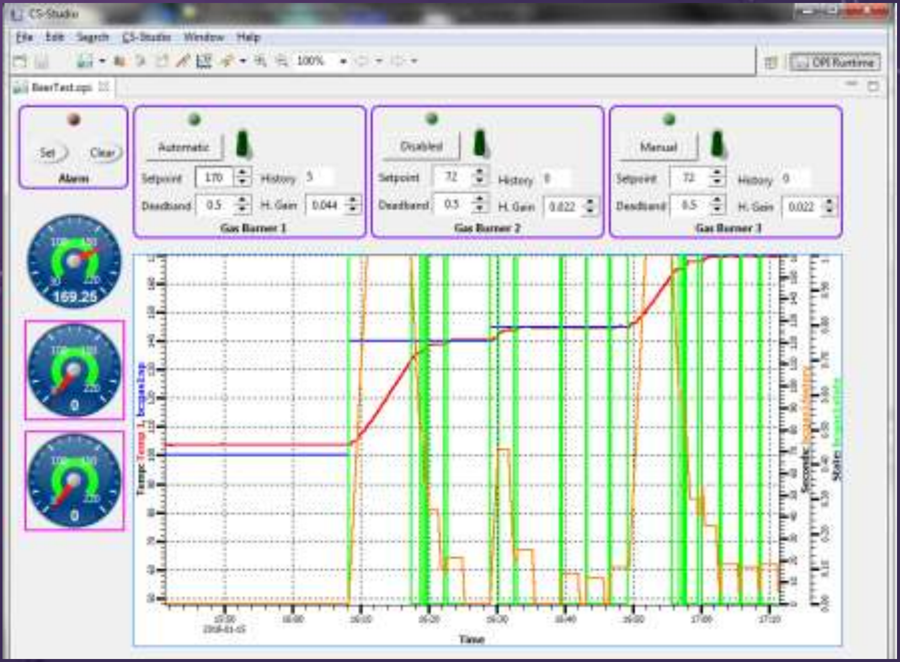
Gas Burner 1: Automatic, Setpoint 120, Deadband 0.5

Gas Burner 1: Disabled, Setpoint 72, Deadband 0.5

Gas Burner 1: Disabled, Setpoint 72, Deadband 0.5

Correcting Overshoot

- To correct for hysteresis, store burner history, e.g. burner was on X seconds out of the last 120 seconds.
- Operator specifies Overshoot, in degrees F, based on observation.
- Turn Burner On if Temp + (History * Overshoot / 120) < Setpoint – Deadband
- Turn Burner Off if Temp + (History * Overshoot / 120) > Setpoint
- Achieves +/- 0.5 F accuracy



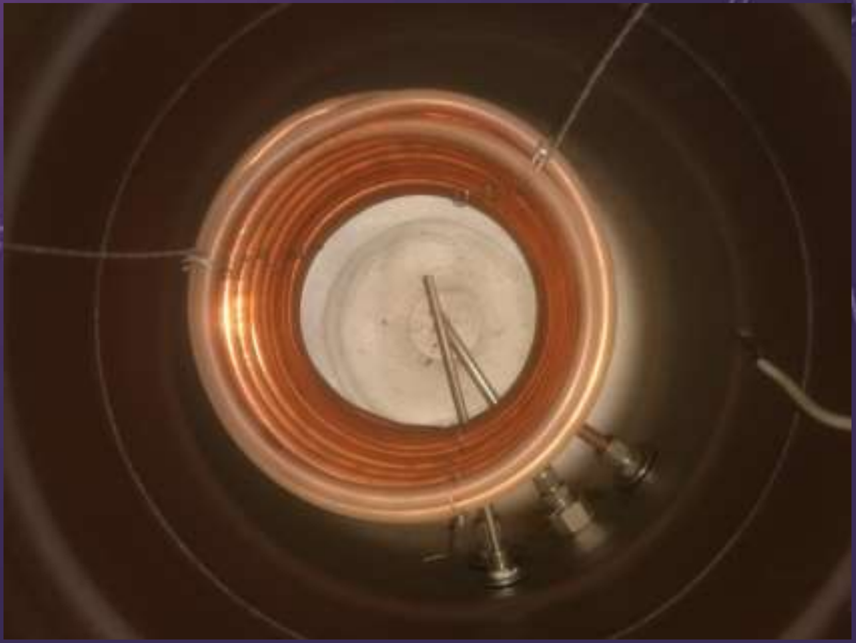
Gas Burner 1: Automatic; Setpoint 170; History 5; Deadband 0.5; H. Gain 0.044

Gas Burner 2: Disabled; Setpoint 72; History 0; Deadband 0.5; H. Gain 0.022

Gas Burner 3: Manual; Setpoint 72; History 0; Deadband 0.5; H. Gain 0.022

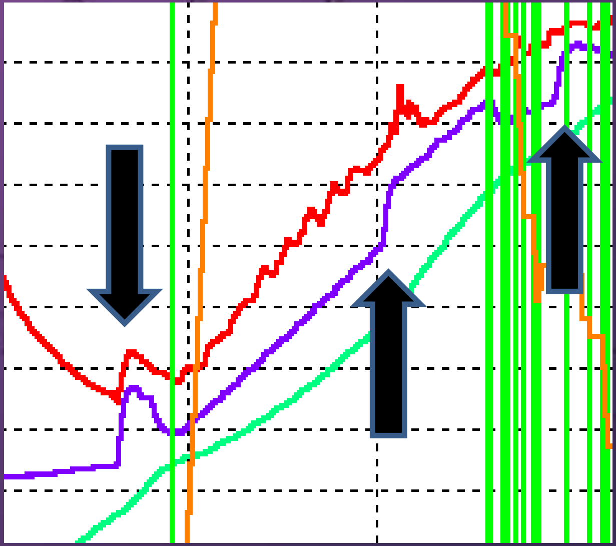
The Heat Exchanger Problem

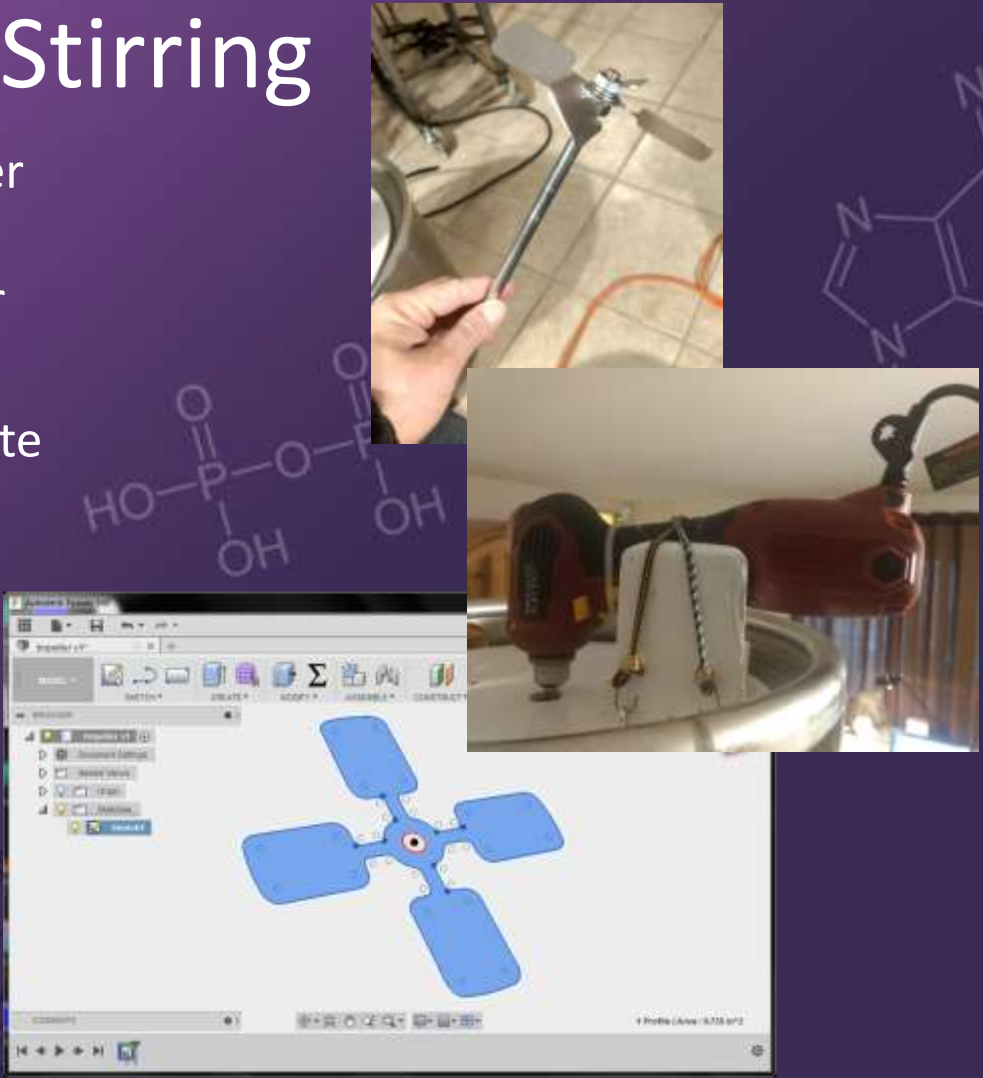
- We measure three temperatures:
  - The grain mash in the mash tun
  - The water in the heat exchanger (directly controlled by the propane burner)
  - The output of the heat exchanger (what we want to control)
- We have a desired setpoint for the heat exchanger output
- We need to calculate a dynamic setpoint for the heat exchanger water, which the propane burner control algorithm will use.



Improving Heat Exchanger Efficiency via Stirring

- We observed the heat exchanger output was closer to the heat exchange water when the water was stirred.
- We added an impeller to circulate water in the heat exchanger.





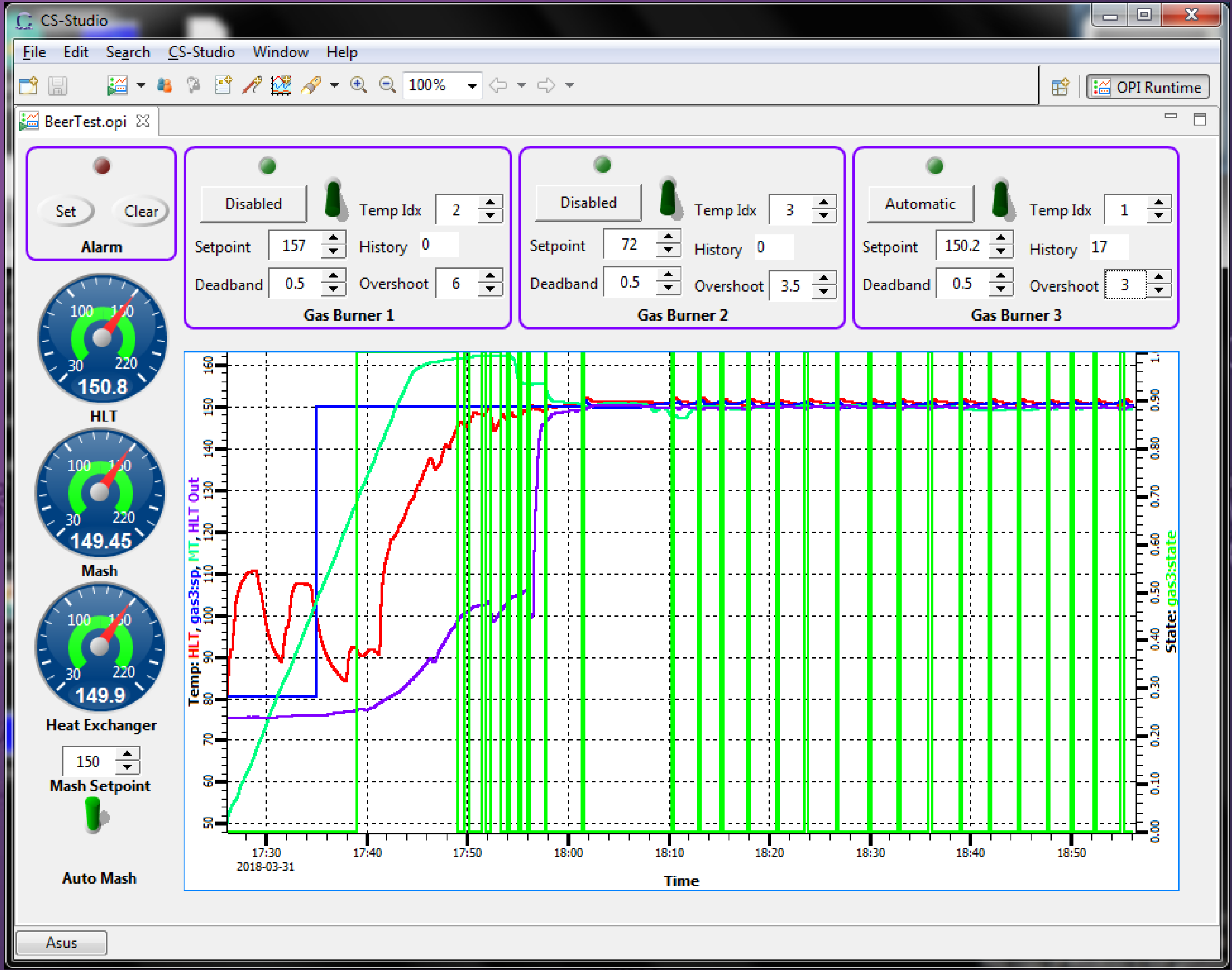
calcout -> Good Beer!

- Experimentation with circulating water showed that:
  - Optimum dynamic heat exchanger temperature setpoint is the desired temperature plus twice the error.
  - E.g. if I want 155 F and the heat exchanger output is 154, the propane burner should use a 157 F setpoint.
- 2 * Error should be capped at 10 F so the heat exchanger setpoint won’t be dangerously high.
- This dynamic functionality is only used during mashing.
  - Preheating before adding the grain, and sparging afterwards, use static heat exchanger setpoints.
  - This functionality requires an on/off switch.
- Success! +/- 0.5 F heat exchanger output is achievable while brewing beer!

```
record(ao, "$(P)mash:sp") {
    field(DESC, "Mash Temp Setpoint")
    field(EGU, "F")
    field(HOPR, "190")
    field(LOPR, "30")
    field(PREC, "2")
    field(PINI, "YES")
    field(VAL, "155")
}
record(bo, "$(P)mash:auto") {
   field(DESC, "Mash Auto Setpoint")
    field(ZNAM, "Off")
    field(ONAM, "On")
    field(PINI, "YES")
    field(VAL, "0")
}
record(calcout, "$(P)mash:calc") {
    field(INPA, "$(P)mash:sp CP")
    field(INPB, "$(P)temp3:temp CP")
    field(INPC, "$(P)mash:auto CP")
    field(CALC, "C")
    field(OOPT, "When Non-zero")
    field(DOPT, "Use OCAL")
    field(OCAL, "A+MIN((A-B)*2,10)")
    field(OUT, "$(P)gas3:sp PP")
}
```

Brewing an IOC IPA

Brewing an IOC IPA



Gas Burner 1
Disabled
Temp Idx 2
Setpoint 157
History 0
Deadband 0.5
Overshoot 6

Gas Burner 2
Disabled
Temp Idx 3
Setpoint 72
History 0
Deadband 0.5
Overshoot 3.5

Gas Burner 3
Automatic
Temp Idx 1
Setpoint 150.2
History 17
Deadband 0.5
Overshoot 3

HLT 150.8

Mash 149.45

Heat Exchanger 149.9

Mash Setpoint 150

Auto Mash

6/14/2018

Analytics Lounge

Next Steps

- Hardware:
  - Complete valve automation
  - Fabricate a Boil Kettle
  - Calibrated water fill via flow rate sensors
  - Fluid level sensors
  - Stepper motor control of propane valve

- Software:
  - Scripting, ramp/soak (SNL? Bluesky?)
  - Archiver Appliance? WebOPI? iPad control?
  - Touch screen on the Pi? (PyDM?)
  - Compare results with beer design software and quantify repeatability

EPICS and the Maker Ecosystem

- EPICS solves the same problems for makers and physicists, even if the hardware being controlled is vastly different
- Both communities strongly value open source, customizable ecosystems
- Considerable crossover (Pi, Arduino) with the common maker hardware / software stack
- Both communities compliment each other with different perspectives
- Physics and the maker movement strive to achieve the audacious!

EPICS and Beer! Any questions?