Emil Gora
Mr. Feldkamp
CSI English 12
1 March 2021

Profiling Espresso Shots Through Control Systems

Have you ever noticed how a cup of homemade coffee brewed one day can be starkly different, for the better or worse, on another day? Many home espresso and coffee machines are limited by their hardware. In my paper, I will explore temperature, pressure, and flow control algorithms and how data collected from them can be used to re-create or perfect an espresso shot; in addition I will explore how software can be used to compensate for hardware inconsistencies and discuss how sensor technologies can be applied to modern microcontrollers. I will use this research for my Capstone project in order to design, code, and create a controller for a Rancilio Silvia espresso machine.

A large controversy around espresso is what a taster considers “perfect.” After all, our taste buds vary down to a genetic level: “The receptors catch the molecules that touch the front of the taste cells. After, they direct a message in the cell to the nerve endings around the cells. The different structure that everyone has come from their genes,” (Walker). Though taste varies, it is still possible to objectively judge espresso based on technical details such as “Entry water pressure, Percolation time, Exit temperature of water from the unit and Millilitres in the cup,” (Odello). Small imperfections can be prevented with a better understanding of how a machine works and is adjusted.

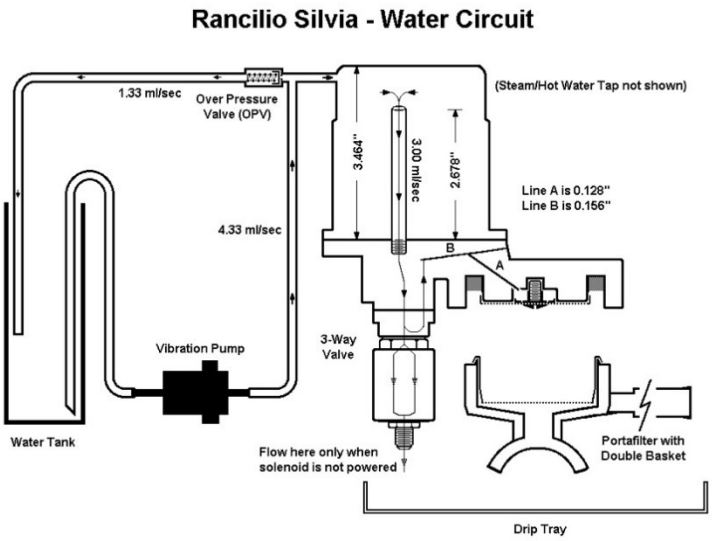


Figure 1. (Svendson)

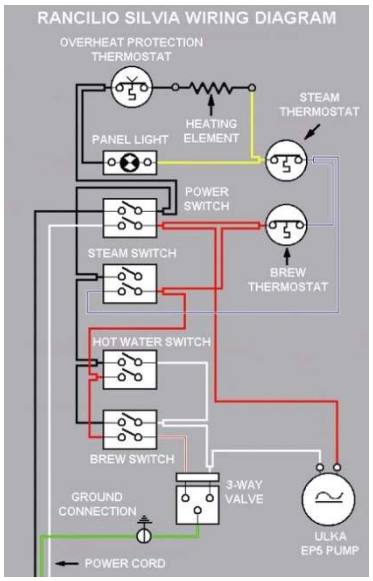


Figure 2. (Svendson)

Before changing stock components, one should understand the role of each part in an espresso machine. By observing the path in figure 1, one can dissect how this machine's specific single boiler works: water starts cold in the tank, flows through the vibration pump, past (or through) the OPV, into the boiler, through the solenoid valve, through grouphead channel A, through grouphead channel B, through the shower screen, through the ground coffee, through the portafilter's basket, and finally into the cup (or drip tray.) The cold water entering the boiler creates pressure, which forces the heated water through the parts after the boiler. This flow path excludes the steam wand, since closing it entirely would prevent the boiler from releasing any pressure. In this stock setup, the pressure is purely controlled by the point at which the OPV opens, which is just over 9 bars (project caffè!). On a side note, it is important to mention that when choosing new parts to add to an espresso machine it is crucial that the parts are

suitable for potable water. Potable water is water considered safe enough to drink, based on standards set by agencies/foundations such as the NSF, or National Science Foundation. The NSF has created varying ANSI, or American National Standards Institute, standards for water systems that define what materials are used in certain parts. OEMs (or Original Equipment Manufacturers) may have been able to pass unsafe machines with parts containing lead in the past, but new NSF guidelines have been put in place to restrict the use of hazardous materials such as lead (Teahan).

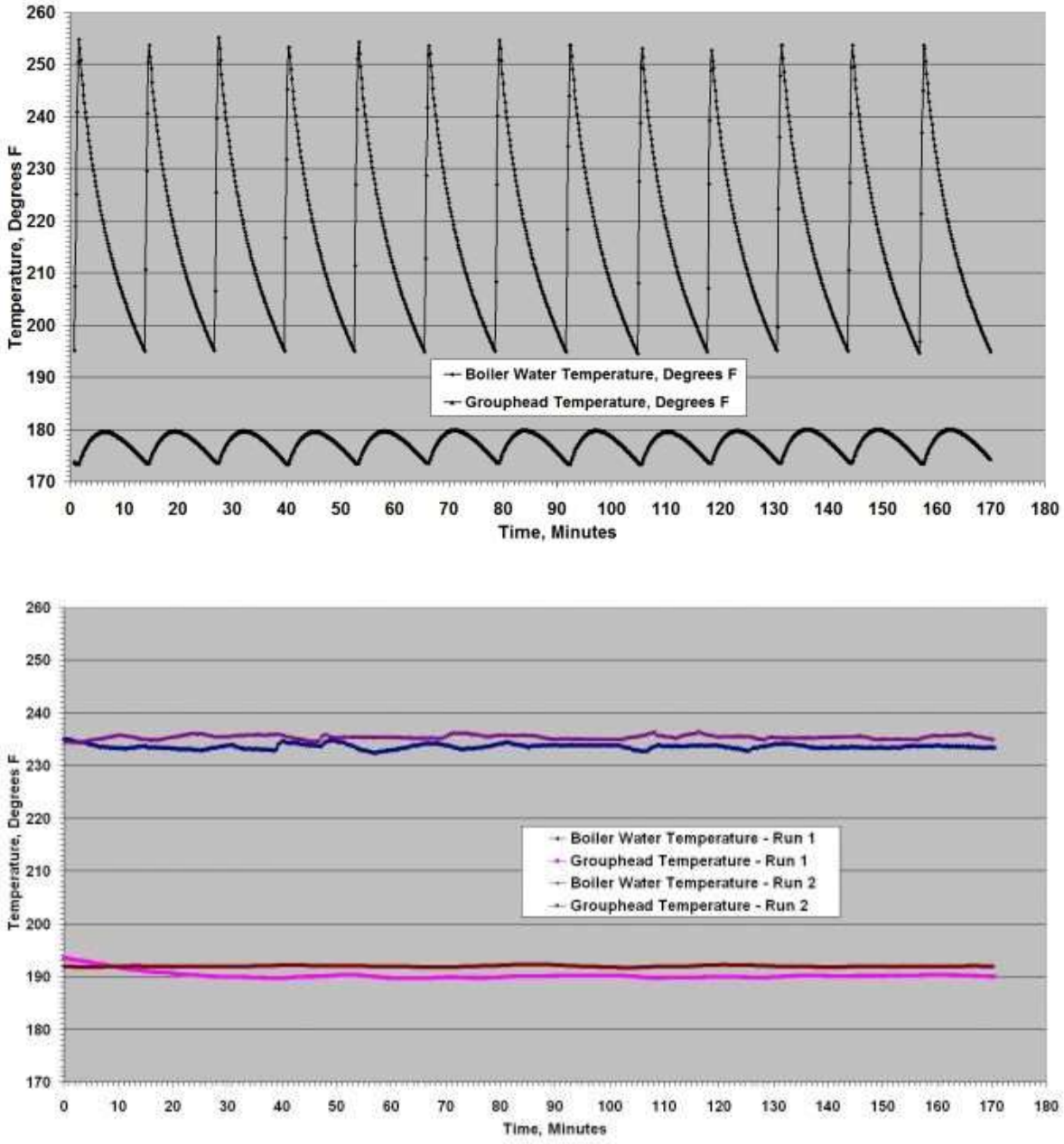
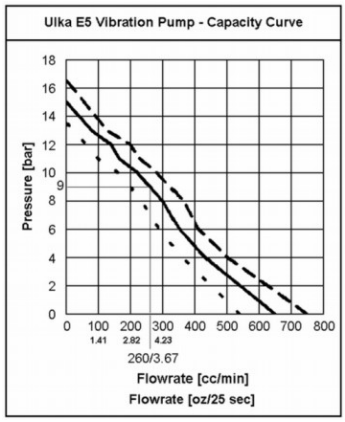


Figure 3. (Svendson)

Seen in the top graph of Figure 3, the stock thermostat (part of the wiring used in Figure 2) on a Rancilio Silvia cannot hold a consistent temperature, which is referred to as hysteresis. A traditional thermostat is a strip of metal inside some form of casing, that moves depending on the amount of heat it is exposed to. Though they are cost effective, traditional thermostats can

be inaccurate and produces inconsistent results. The peaks are roughly 250deg Fahrenheit and the valleys are roughly 200deg Fahrenheit, which is a significantly large range of temperatures for the boiler that could potentially make a shot sour or underdeveloped. The grouphead temperature range is less dramatic but is not ideal either. In contrast, a thermocouple is a more precise measuring instrument because it contains dissimilar metals that deform as heat changes, which generates a low voltage, but precise, output. An amplifier can be used to convert a low voltage signal to a logic-level analog signal or to a digital signal such as serial. On the lower graph the boiler is being controlled by a PID loop, using a thermocouple and a solid-state relay (Svendson). By using this PID loop, a temperature input, and a method of switching the heater on and off at a high frequency, a stable temperature can be held (project caffè!).

> A formula uses all three terms to control equipment to the desired results. Tuning a PID Loop sets optimal parameters in each variable to get an ideal response from a control system. There are different methods of tuning, but a common one is “guess and check.” That is, set variables by experience with that equipment and then check the results. Is the equipment performing as desired? If not, tune some more. (Soper).



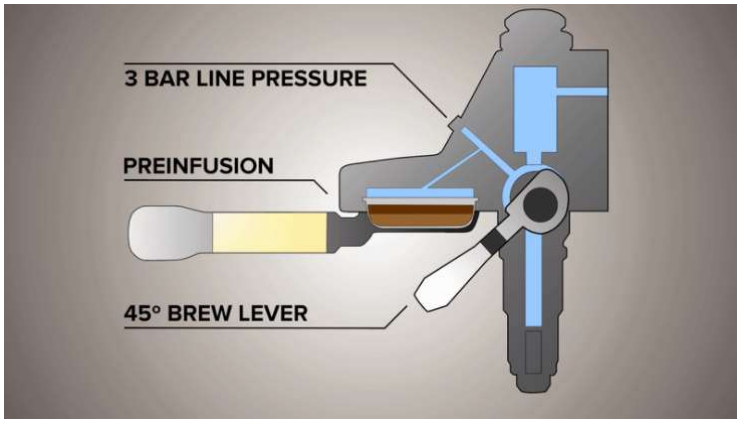


Figure 4. (Svendson) Figure 5. (Kelso)

Controlling pressure is fundamentally similar to controlling a heater. By adding a pressure transducer, a flow meter, a scale, necessary plumbing, and hardware the machine's outputted data can be recorded into a database and profiles can be created with that data. In the case of ESPHome, most data is stored in or transferred to an SQL database. A pressure profile, for instance, is essentially a user-defined set of values that adjusts a level of pressure that should be maintained and for how long it should be maintained.

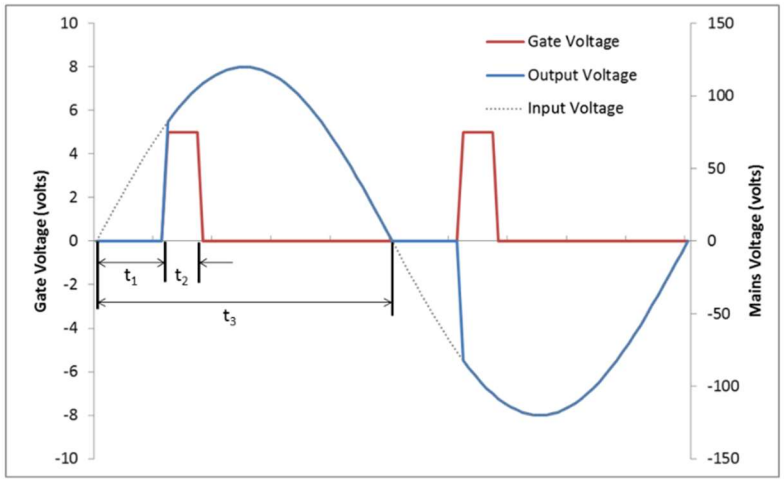


Figure 6. (Arduino)

Though Figure 5 uses an E61 grouphead as an example for pre-infusion, it is also conceptually similar to how phase angle control functions interact with a vibratory pump (Kelso). An E61 grouphead has a rotating Cam that can control the amount of pressure passing through, in comparison to a circuit that controls an AC waveform. Figure 6 demonstrates how phase angle control is visualized. This concept can be applied to switching components such as TRIACs and zero cross detection opto-isolators. A TRIAC is essentially a semiconductor used to switch an AC signal on or off and an opto-isolator is an integrated circuit that creates a galvanic separation between a logic level input circuit and an AC source. This separation is achieved by using light to control a light affected transistor inside of an IC package.

> Once a zero crossing is detected, the TRIAC remains off for a controlled amount of time (t1). The longer this time is, the less power the AC circuit receives. Once the “off-time”, t1 has elapsed, the microcontroller turns on the TRIAC by applying a voltage to the gate (shown in red). Once turned on, the TRIAC will remain on even after the gate voltage has been removed. It will turn off if the gate voltage is zero the next time the AC wave crosses zero. (Arduino)

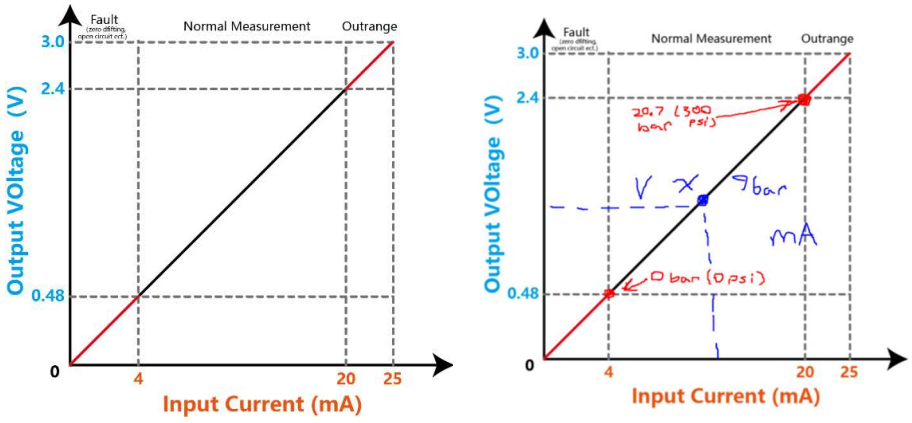


Figure 7. (DFRobot)

Pressure transducers convert pressure into a signal by sealing a membrane/diaphragm against a fitting that can be installed in-line with plumbing. Though the ESP32 has analog inputs and ESPHome has an analog sensor by default, without modification these are only compatible with ratiometric voltage pressure transducers. Ratiometric current pressure gauges require an extra circuit in order for the microcontroller to process the signal. The chart in figure 7 shows the relationship between the analog output of a DFRobot TP5551 module and a current based ratiometric transducer. An analog output could be used with nearly any Arduino since most Arduinos have built-in digital to analog converters. On the second chart, I annotated corresponding maximum and minimum current values of an IPSU-GP300-5M12 transducer to serve as an example. Increasing or decreasing the phase angle, proportional to point X, would be contained in a loop (DFRobot). The proportion would also be defined by a user-profile, by changing points on a graph, or by writing code specifying how they would want to pre-infuse a shot. Flowmeters are generally enclosed in metal casings and utilize a hall-effect sensor to track how many milliliters of water have passed though the machine. Prior to understanding the real flow rate of a sensor, one must record the amount of pulses created by a flow meter while keeping track of the amount of water passing through over a period of time. Placement of a flow sensor can be deceiving because of the OPV in Figure 1. If placed after the OPV, a flow meter has a better representation of the true amount of water passing through the boiler since there isn't a probability of loss through the OPV. This might seem straightforward, but it creates another issue of space management and fitting leakage (project caffè!).

Since simultaneously having many digital inputs, PID loops, phase angle calculations, and data queries over a network is resource intensive, a standard Arduino Uno/Micro would

not be an ideal board to design a controller around. In addition to having more processing power, the ESP32 stands out from many other microcontroller development boards for its features. It supports clock speeds up to 160Mhz, built-in Wi-Fi and Bluetooth interfaces, 520Kb of random access memory, and up to 10 touch sensors. Even with all these features, the board manages to have a footprint smaller than that of an Arduino Uno or Arduino R3 and isn't much more expensive, relatively speaking (David). Since most control concepts previously mentioned can be tedious to create from scratch in a coding language such as C/C++, a demand for a scalable, yet user-friendly framework for creating custom firmware became apparent to developer communities. ESPHome, created for ESP8266 and ESP32 microcontrollers, fills in this need by using configuration files to structure code rather than over-complicated sketches. ESPHome uses a compiler to generate firmware files using libraries, which is similar to the Arduino compiler, but it does so in a different way. A user writes their configuration using a YAML (an acronym for "YAML Ain't Markup Language") format file which uses IDs to contain data, rather than variables. The assigning of variables, integers, floats, Booleans, and just about every other primitive data type happens through a backend, which is considered data-serialization. This makes YAML suitable for a modular system that is easily adaptable to hardware changes. With this scalability, users are able to easily create custom libraries that call upon the ESPHome library within them (Winter). Outputting data via a user interface is another challenge for developers, as a frontend, such as HomeAssistant can only handle a finite number of packets from an ESP32 at a given time. By default, the history graph card on HomeAssistant only supports showing 1 entity/value across a given number of hours, not seconds, which only allows a user to view long-term statistics. In this case, it may be necessary to write a more

precise plotter that can graph multiple values at once, can display a short-term timespan, and can update faster than the default card. In addition, recording and exporting the graph would be a convenient feature for future reference (Winter).

Temperature sensors, such as the TSic 306, could be integrated into ESPHome using existing libraries that are written for C/C++ Arduino code. This could be achieved with the modification of pin mappings and variables, so that an ESPHome configuration can call upon the library. In addition, the Bluetooth low energy functionality of a Skale II smart scale would pair well with ESPHome because of built-in libraries that can read weight values over specified memory addresses. Placing a scale when tracking the weight of a shot is also known as "gravimetric profiling." Point one gram accuracy scales are ideal for espresso since more subtle changes can be tracked/corrected. I mention these 2 integrations because they are also used in a less open-source project named ito!, which is also an Atmel-based controller (project caffè!). While the binaries are open-source for public use and modification, its source code is not available to the public and firmware/hardware manuals are subject to copyright (project caffè!). This makes open-sourcing appealing to me as a developer because anyone can contribute to code online and learn everything behind the scenes.

In conclusion, by synthesizing what I have learned through my research, I now have a deeper understanding of every component in an espresso machine. I have discussed possible applications of different sensors and components, which can be used to solve my original dilemma of pulling inconsistent espresso shots.

Works Cited

Arduino. "AC Phase Control." n.d. *Arduino Playground.* <https://playground.arduino.cc/Main/ACPhaseControl/>.

David, Christopher. *Diy I0T*. July 2019. <https://diyi0t.com/technical-datasheet-microcontroller-comparison/>.

DFRobot. *Analog Current to Voltage Converter for 4~20mA*. n.d. <https://wiki.dfrobot.com/Gravity__Analog_Current_to_Voltage_Converter_for_4~20mA_Application__SKU_SEN0262>.

Kelso, Charles. "What is Pre-Infusion?" 9 May 2019. *Clive Coffee.* <https://clivecoffee.com/blogs/learn/what-is-pre-infusion>.

Odello, Luigi. "The Certified Italian Espresso and Cappuccino." December 2006. *Instituto Espresso Italiano.* <http://www.espressoitaliano.org/files/File/istituzionale_inei_hq_en.pdf>.

project caffè! *leva! 1.9 Firmware Manual*. 2021. <http://projectcaffe.bplaced.net/files/leva!19-ito.rar>.

—. "leva! 1.9 Hardware Manual." 16 November 2020. <http://projectcaffe.bplaced.net/files/leva!19-ito.rar>.

Soper, Mike. "Simple Examples of PID Control." 27 July 2017. *Setra.* <https://www.setra.com/blog/what-is-a-pid-loop>.

Svendson, Eric. 24 September 2007. <http://users.rcn.com/erics/Rancilio%20Silvia/Sil_Watr.jpg>.

—. 5 April 2007. <http://users.rcn.com/erics/Rancilio%20Silvia/Sil_Elec.jpg>.

—. 5 April 2007. <http://users.rcn.com/erics/Rancilio%20Silvia/ULKA5_21.jpg>.

—. 15 August 2007. <https://www.home-barista.com/espresso-machines/rancilio-silvia-performance-with-without-pid-t4691.html>.

Teahan, Michael. "Get the Lead Out." 27 September 2019. <https://coffeetechniciansguild.org/blog/2019/9/23/leaded-coffee>.

Walker, Courtney M. *Why Are Everyone's Taste Buds Different?* 22 October 2015. <https://sites.psu.edu/siowfa15/2015/10/22/why-are-everyones-taste-buds-different/>.

Winter, Otto. *ESPHome Wiki*. n.d. <https://esphome.io/>.