1 Purpose

Documentation for Simple Brew Control Firmware, which is a microcontroller program that can be configured as thermostat, boil controller, kegerator thermostat, or fermentation controller.

3. Index

List of Figures

2 Objective

Temperature control and PWM control of heating elements is a common need in beer brewing and sous vide. This is a simple microcontroller firmware that can fulfil these roles and possibly replace expensive special-purpose PID controllers and temperature controls. The firmware also functions as a basic PWM generator and should be useful for other temperature control applications that need selectable open-loop control as well as closed-loop thermostatic control in the same unit.

3 Microcontroller hardware

The microcontroller used is an ATMEGA328 8-bit AVR© microcontroller which is widely available for about 3 USFRN in convenient packages. It was selected for its availability in my toolbox. The firmware would run on other AVR microcontrollers with minimal modification. The microcontroller is powered by 5V and runs at 1MHz, which is how it's configured out-of-the-box. No external crystal is needed, just apply 5V on the Vcc pin and pull up RESET. Refer to figure 7 for microcontroller connections. Note that if you use a 5V cell-phone power supply, you don't need any voltage regulator.

Flashing this firmware to the microcontroller will require an in-system-programmer such as the AVRISPV2 or USBTinyISP. Instructions on programming AVR microcontrollers is outside the scope of this document. All the cool kids are using Arduino nowadays, and you may be able to pretend this is an Arduino program and try to flash it with the Arduino IDE and it may Just Work. You are on your own on the Arduino front. If you ask nicely, I may be able to program one for you or send you a programmed chip for a nominal fee.

4 Configuration Possibilities

The firmware is very flexible. It can be a PWM source, thermostat, or full-featured brew controller. You can make an increasingly featureful system by hooking up more hardware.

PWM

Simply hook up two resistors to the BOIL analog input (PC0) as a voltage divider (figure 4). The system will then toggle the relay output PB2 at a duty cycle proportional to the voltage applied. Refer to table 3 and figure 4 for resistor values. In this mode you essentially have a pulse-width-modulation (PWM) generator that is similar to all the 555-timer-based PWM circuits out there. If all you need to do is knock down a heating element's wattage, this might be all you need.

If you want variable output, just use a potentiometer instead of 2 resistors–then, the firmware is a varible PWM generator. You could use the analog input for feedback, if you so desire.

Boil Override Switches

You can add more options to your PWM controller by hooking two switches (or one 3-position "center-off" switch) from ground the BOIL and WFO digital inputs on PB5 and PB3, respectively. This is the same as above–it's still a PWM generator–only now there are 2 additional options: WFO (Water Fully On) and BOIL (fixed 60% by default). Turning these switches on overrides the main PWM setting on PC0. This allows you to get your knob set where you want it, but still have an easy override to 100% and (by default) 60%. This could be a good configuration for a hot-wire foam-cutter.

Thermostat Functionality

You can use analog inputs PC1 and PC2 to create a thermostat. Input PC1 reads the temperature sensor, and PC2 reads a temperature setpoint knob. In this mode, the firmware turns the output on only if the temperature (voltage on PC1) is below temperature setpoint (voltage on PC2). If you want a "cool" mode thermostat, such as for a kegerator, just swap the PC2 and PC1 inputs. Note that once the thermostat commands the output ON, the duty cycle of the output is still whatever is set by PC0. You need to connect PC0 to +5V, resistors, or a pot depending if you want the output % to be full, partial, or adjustable. For a kegerator, you definitely want to pull PC0 up to 5V.

5 How I use it

For brewing, I use the full configuration. I use the thermostat functionality with the power knob turned all the way up to heat my mash water to strike temperature while I do something else. After I dough-in my grain, I again use the thermostat mode to maintain mash temperatures, only I turn the knob down to about 30% for more gentle heat to prevent scorching. After the mash, I disconnect the temp probe (you don't need to, though) and kick the switch to WFO until I have a good boil, then kick it to BOIL for a very consistent and repeatable boil with a predictable rate of boil-off.

6 Stuff you didn't want to know

The PWM duty cycle is about 1Hz by default. This is good for most brewing/cooking applications driven by mains electricity. To increase it to 4Hz, connect PB6 to ground. To increase it to 240Hz (maybe for hot-wire cutting or dimming LEDs), connect PB7 to ground. For 2kHz, which is good for small DC motors, connect PB1 to ground. However, the 2kHz setting may decrease accuracy of analog readings.

Kegerator thermostats typically have a "compressor delay" feature to protect the compressor from fast-cycling. To enable a compressor delay feature, connect PB0 to ground.

7 Perhipheral hardware options

Input Item	AVR Pin	Voltage details	Function notes
Boil Switch	PB3	GND=BOIL	Override duty cycle for boiling
WFO Switch	PB4	GND=WFO	Override duty cycle to 100%
Temp probe	PC1	0V-5V analog in	LM335
BOIL pot	PC0	0V-5V analog in	Heating element power
TEMP pot	PC2	0V-5V analog in	Temp setpoint

Figure 1: Table Of Inputs

Output Item	AVR Pin	Voltage details	Function notes
Blinkenled	PB5	LED+resistor	Blinks
SSR	PB2	Hook to SSR	5V=heater on

Figure 2: Table Of Outputs

R1	R2	Voltage	Duty Cycle
0	∞	5V	100%
1000	4000	4V	80%
2200	2200	2.5V	50%
4000	1000	1V	20%
∞	0	0V	0%

Figure 3: Resistor values for various fixed PWM duty cycles

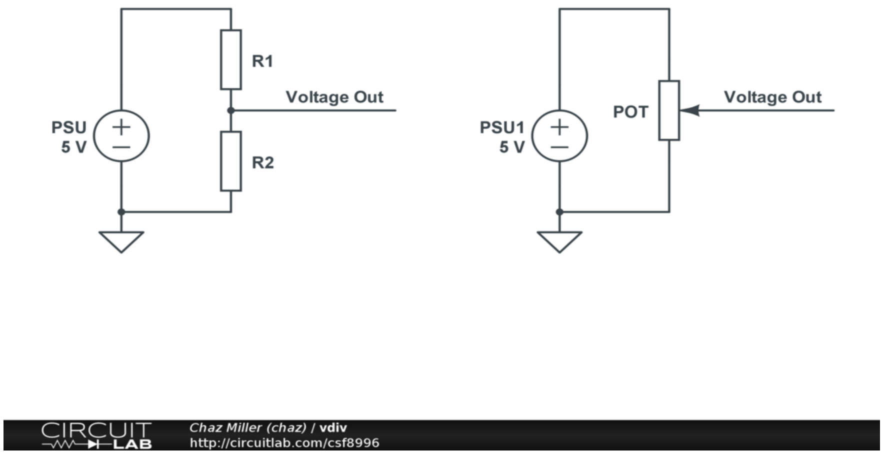

Figure 4: Fixed (left) and variable (right) voltage dividers

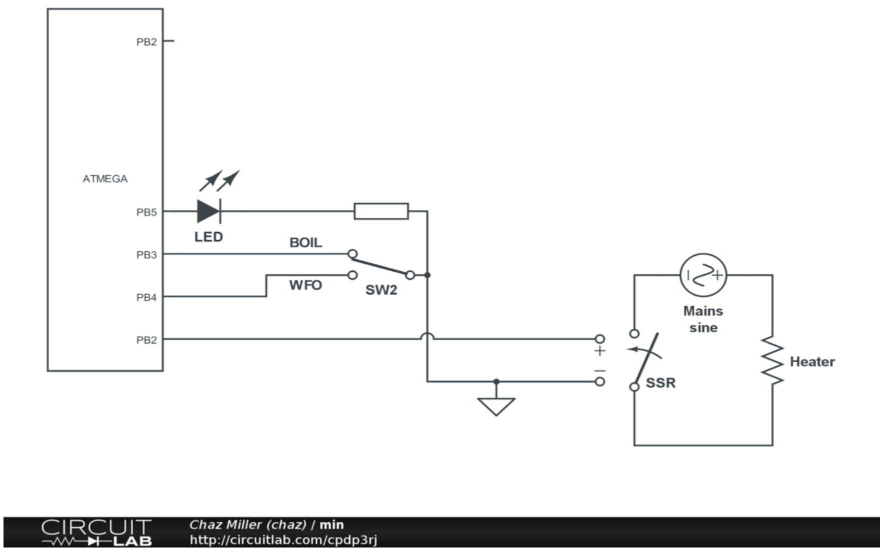

Figure 5: Example Minimal System

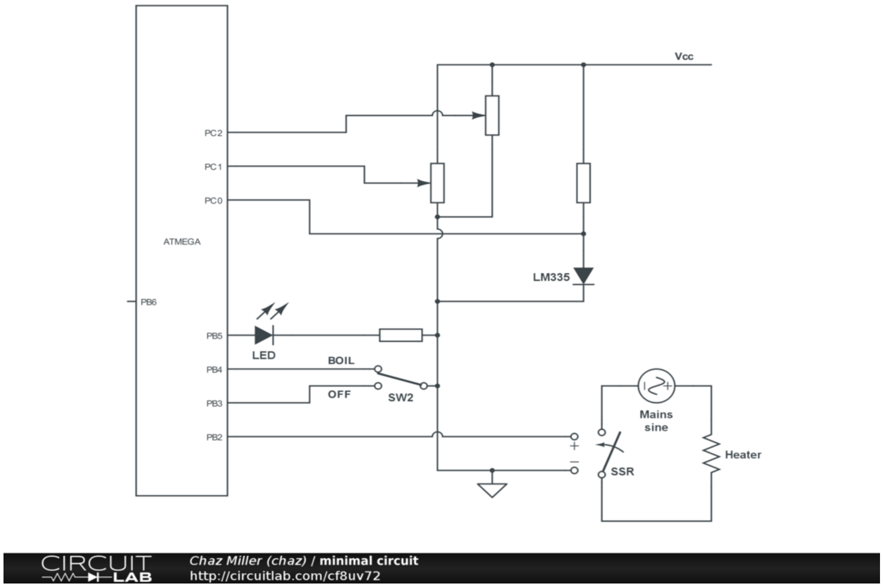

Figure 6: Example Complete System

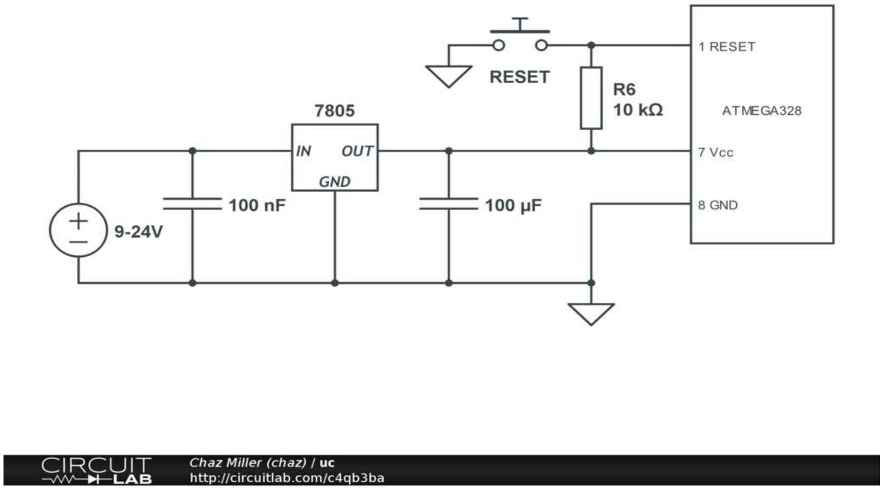

Figure 7: Microcontroller connections

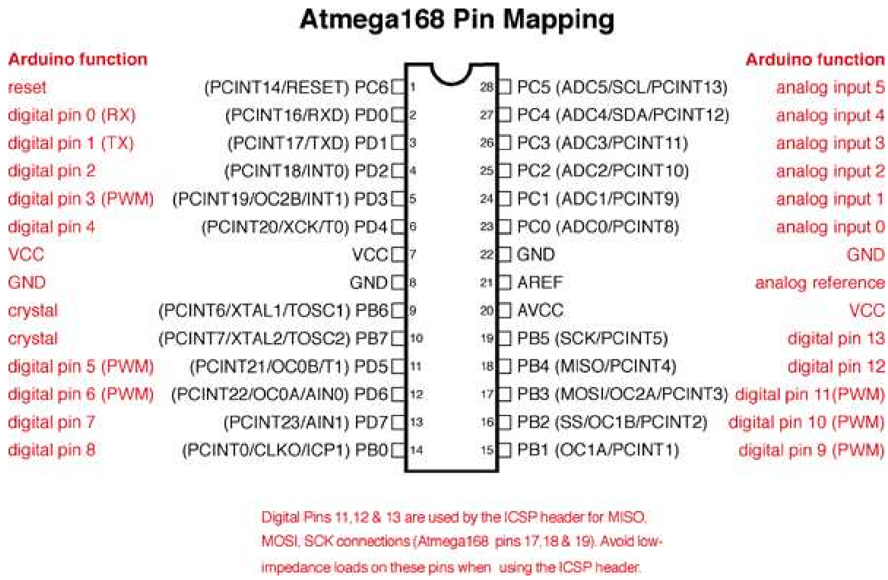

Figure 8: Pinout for ATMEGA168/328