bcc.py is a python program written exclusively for the Beaglebone Black™ micro-controller with the intention of controlling fermentation temperatures in the production of alcoholic beverages. It is assumed the user is well versed in fermenting and has at their disposal either a refrigerator or freezer and some relay system to turn the voltages on and off to the device. An optional relay can also be used to control a heating source in the event temperatures drop below the recommended fermentation temperature.

Brew Chamber Controller

Instruction Manual

Table of Contents

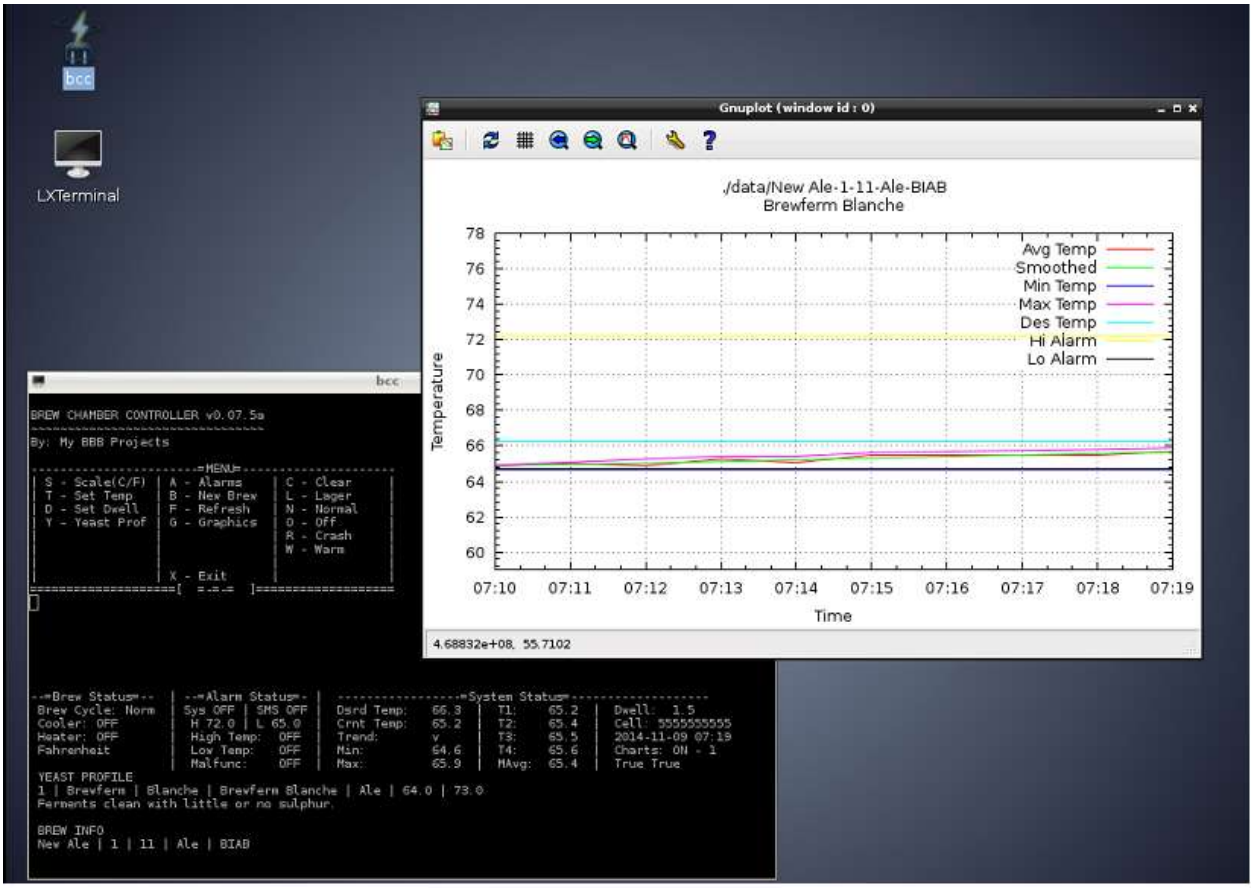

Fig 1 - Version 0.07.05a running in Debian on a Beaglebone Black Rev C showing a graph of temperatures.

1 Introduction

1.1 What is bcc.py?

bcc.py is a text based program written in python to run exclusively on a Beaglebone Black (BBB) Rev C with a 4GB eMMC micro-controller. It may run on earlier versions but space will be a premium on a 2GM eMMC. If you use an earlier version of BBB you also will need to prune the files in data directory once in a while to keep the file system from getting full.

The basic function of bcc.py is to determine if the current temperatures in the fermentation chamber and/or of the wort under fermentation are within tolerance of the desired fermentation temperature (based on the yeast strain) and to turn on a cooling or heating source to bring it back within tolerance. Other features built on top of that basic function are ancillary and not essential but make the program more interesting and informative for the user.

bcc.py in its current form does not automate the fermentation process but is expected to if the user has the necessary sensors. Once that functionality has been built into bcc.py this paragraph will be removed.

1.2 Why the Beaglebone Black?

I choose the BBB because of its increased computing power and available memory over other similar micro-controllers, any of which can and have been used to create a temperature controller. If no LCD screen will be used in the final product then any of the other micro-controllers would be a better option based on price. However if the final product is to have a touch screen of some sorts to provide user input, the price of the BBB touch screens are less expensive resulting in a less expensive project.

1.3 Why Python?

I wanted a simple programming language that didn't involve compilers to keep the project simple for the uninitiated. I could have released a compiled C program ready to use but I wanted hobbyists to get involved in their own programming and make the code their own. I felt a compiled language would deter that to some extent. I initially used the perl language when learning how to program and read the IO pins on the BBB. Though there is nothing wrong with perl and I use it for scripting a lot, I didn't feel it was made for this extensive a project. I wanted to be able to use functions and threading and system calls and felt python was better suited for this. Is python a good beginner's language? Hardly not, and I discovered that while learning python myself on this project. But I find python to be a very capable language and can do a lot of things simply once you know how to use it. I'm still learning, so expect the code to go through many changes as I discover better ways to accomplish things in python.

I should also mention that the code is written and commented so the uninitiated can follow along and understand what the program is doing. I could easily have reduced the number of variables used in the program and written the code more concisely but I feel this would have obfuscated the code to where it would not be so easy to understand what the code was doing.

2 Installation

Some knowledge of Unix/Linux administration is required and not covered in this documentation. There is extensive help online on how to be a Unix/Linux administrator. The following is a guide on how to install bcc.py and the required programs on the BBB.

2.1 Create bcc user account:

- ✓ ssh into BBB as root
- ✓ adduser bcc
- ✓ password (whatever you want)_____
- ✓ default the rest of the questions

2.2 Install prerequisites on Beaglebone Black (BBB):

- ✓ ntpdate pool.ntp.org
- ✓ dpkg-reconfigure tzdata (select your timezone)
- ✓ pico /etc/adjtime (change UTC to LOCAL)
- ✓ apt-get update
- ✓ apt-get upgrade
- ✓ pip install --upgrade Adafruit_BBIO
- ✓ apt-get install gnuplot-x11
- ✓ apt-get install gedit (**optional** – better editor for code editing)

2.3 Install bcc on Beaglebone Black (BBB):

- ✓ ssh into BBB as root
- ✓ cd /usr/local
- ✓ git clone https://github.com/cyberlord8/bcc
- ✓ chown -R bcc ./bcc
- ✓ cd bcc
- ✓ chmod +x ./bcc.py
- ✓ chmod +x ./su-bcc.run
- ✓ visudo
- ✓ Add bcc below the debian entry (follow the debian entry format)
- ✓ pico /etc/group
- ✓ add bcc to sudo and staff groups
- ✓ su bc**c**
- ✓ mkdir ./data
- ✓ mkdir ~/Desktop
- ✓ cp bcc.desktop ~/Desktop/

3 Running bcc.py

3.1 To run bcc:

- ✓ ssh into BBB as bcc
- ✓ vncserver :1
- ✓ vnc into BBB as bcc
- ✓ double click the bcc icon on the desktop
- ✓ if this is the first time running bcc.py then enter your cell number (or 5555555555)
- ✓ optionally, clean up the terminal window (this affects all terminal windows)
- ✓ right click in the terminal window
- ✓ click on preferences
- ✓ click on the display tab
- ✓ hide the scroll bar
- ✓ hide the menu bar

3.2 Testing bcc:

3.2.1 Brew Session

- ✓ Type B <enter> to start a new brew session
- ✓ Enter a brew session name (usually the name of the brew)
- ✓ Enter a batch number (increment the batch number if you are remaking a brew)
- ✓ Enter the batch size (in gallons or liters)
- ✓ Enter the brew style (Ale, Lager, etc.)
- ✓ Enter the brew method (Brew in a Bag BIAB, AG, Extract etc.)
- ✓ Enter a yeast strain ID from the Yeast Profile database
- ✓ Type N to start the brew session

3.2.2 Graphing

- ✓ Type G to set graphing parameters
- ✓ Type yes to enable or no to disable graphing
- ✓ If enabling graphing enter the refresh interval (1 for testing)
- ✓ If the brew session is not running type N or W to start it. The chart should appear once 2 data points have been logged (two minutes give or take if you set the refresh interval to 1)

4 Operation

4.1 The Menu System

4.1.1 S – Scale(C/F)

This menu option switches between Fahrenheit and Celsius. Values stored in the Yeast Strain.csv file (discussed later) are stored as Fahrenheit but bcc takes this into consideration when selecting a yeast strain.

4.1.2 T – Set Temp

This menu option allows the user to manually set a desired brew temperature. bcc.py uses the temperatures in the Yeast Strains.csv file to automatically set the normal and warm brew temperatures. If you need to adjust the desired temperature from these automatically calculated temperatures then this is the way to do it. It's important to know that selecting a new (or the same yeast strain) with the Y menu option the normal and warm temperatures will be reset and the desired temperature adjusted accordingly.

4.1.3 D – Set Deadband

The deadband setting is how far you want the brew chamber temperature to deviate from the desired temperature. A smaller number means a more precise temperature but more times the compressor or heater kicks on to maintain the temperature. Deadband is used to calculate the min and max alarm temperatures and a large value can skew this calculation up. Recommended setting is 1.5F to 2F and no more than 3F. 3 degrees deadband will result in a 3 degree temperature swing (1.5 degrees above the desired temperature to 1.5 degrees below the desired temperature). You are welcome to use a smaller deadband setting but too low a setting may make the software try to switch the compressor on to soon after it has shut off. While bcc.py has a protection feature that keeps this from happening, it's best to increase the deadband setting so the software doesn't try to turn the compressor on too soon.

4.1.4 Y – Yeast Prof

The Yeast profile menu option allows the user to select a desired yeast strain from a comma separated value (csv) spreadsheet and use the data pertaining to that particular yeast strain. Selecting a new yeast strain or reselecting the same yeast strain will reset the normal, warm, and consequently the desired temperature settings along with the min and max alarm settings.

4.1.5 A – Alarms

The Alarms menu option allows the user to turn the alarm system off altogether or selectively turn the SMS texting feature off by itself. When you initially turn the alarm system you are asked to manually set the min and max alarm temperatures. This is also how you would override the automatically set alarm temperatures when selecting a yeast strain. If you want to use the SMS texting feature you must have entered your phone number when bcc.py was run for the first time. You can change this number by editing bccconfig.py. The SMS feature is set to send no more than 1 text per hour. There is a maximum limit of 3 texts per hour (75 per day actually)

set by the SMS service that bcc uses. However 1 text per hour is probably adequate for what we are trying to achieve.

4.1.6 B – New Brew

The brew info menu option gathers information about the brew that will be used to aid in the automation process when added in a future version. Currently the brew info option asks for Brew Name, Batch Number, Batch Size, Brew Style, and Brew Method. More data will be gathered in future versions to aid in the automation process. Creating a new brew resets a lot of parameters in bcc.py.

Don't forget to start the brew cycle by selecting Warm or Normal menu options after a new brew session is created.

4.1.7 F – Refresh

Since this version of bcc is a text based program running in a terminal sometimes the screen can get messed up. This menu option will clear and redraw the screen. If you resize the terminal window you are likely to mess the display up so it will not format properly. Exiting out of the program and restarting from the desktop icon should fix the terminal window size.

4.1.8 G – Graphics

Turns graphical charting on or off. If on, set the interval to write the data to the database. For details on how to use gnuplot to chart your brew session see paragraph 5.4. A smaller interval will produce larger data files but produce a smoother chart. Too large an interval will display a course chart and may miss a temperature swing (the alarm system will still trigger). Recommended interval is 15 minutes.

4.1.9 C – Clear

The Clear menu option is used to store/cellar/clear the final product. Bottles can be stored in the brew chamber also at cellar temperatures. If you want to store the contents at refrigerated temperatures then use the R – Crash menu option. This value can be adjusted by editing the bccconfig.py settings file.

4.1.10 L – Lager

The lager option is used to maintain a cooler fermenting temperature when lager conditioning a beer. This value can also be adjusted from the bccconfig.py settings file.

4.1.11 N – Normal

The normal brew cycle is the primary cycle used when fermenting. The temperature used is calculated from the variables in the Yeast Strain.csv file. You can manually override the calculated value with the T – Set Temp menu option but remember if you select a new yeast or reselect the same yeast all the temperatures are automatically recalculated.

4.1.12 O – Off

The Off menu option turns the brew cycle off which turns off the cooling and heating functions along with the alarm system. If the brew cycle is turned back on again the system should return to the previous settings except if you selected a different brew cycle from the previous one.

Don't forget to crack the freezer door when the system is off!

4.1.13 R – Crash

The crash menu option drops the chamber temperature which causes the yeast in suspension to stop their activity and drop out of suspension. This cycle should be used when the fermentation has naturally stopped or you are forcing it to stop because a higher FG is desired. If the fermentation is stopped prematurely then steps need to be taken to assure fermentation does not reoccur in the bottle.

4.1.14 W – Warm

The warm cycle is an optional cycle that may be used to help fresh pitched yeast get a good start in fermenting. It is calculated from the variables in the Yeast Strain.csv file. You can manually override this temperature with the T – Set Temp menu option.

4.1.15 X – Exit

The exit menu option stops all functionality of the bcc program, saves the current settings to the bccconfig.py file and exits the program, closing the terminal window. The program can be restarted by double clicking on the bcc desktop icon.

Don't forget to crack the freezer door when the system is off!

4.2 The Display System

4.2.1 Brew Status display

The Brew Status display shows the current brew cycle, whether the cooler or heater is on and whether the Fahrenheit or Celsius scale has been selected. Yellow colored lettering indicates the system is temporarily halted while the bcc program reads temperatures over a certain time period or waits 5 minutes before turning the compressor on again. Once that time period has expired the system is live and ready to operate.

4.2.2 Alarm Status display

The Alarm Status display shows the status of the alarm system. The status of the alarm system as a whole is shown along with the SMS text messaging feature. The high and low alarm temperatures are shown. Also shown are the high temp and low temp alarm statuses. Currently the malfunction alarm has not been written into the software but should be available in a future version once it has been decided what entails a malfunction.

4.2.3 System Status display

The system status display shows the desired temperature, the current temperature, the temperature trend (up/same/down) but still needs some tweaking to make it useful. The lowest and highest temperatures in the brew chamber are indicated by the min and max numbers. The T1 through T4 numbers are the temperature readings taken every 15 seconds for the last minute. The MAvg temperature is the average of the last 4 readings. The Deadband is also shown in this display area.

4.2.4 Yeast Profile display

The Yeast Profile display shows some of the characteristics of the selected yeast from the Yeast Strain/csv file along with a brief description.

4.2.5 Brew Info display

The Brew Info display shows the data entered by the user from the Brew Info menu option. Currently this is for info only and the data is not used by the bcc.py to make any decisions.

4.3 The File System

4.3.1 database.csv

Every 15 minutes or when a menu option is selected the current system status is logged to a database. This database will be used for historical purposes.

4.3.2 Yeast Strains.csv

A database containing a selection of yeasts gathered from the internet. This file can be edited; the only caution is the Yeast ID needs to be sequential with 0 on the header line and 1 as the first yeast strain in the list. You may trim it down to only yeasts you use or add more yeast strains to it.

4.3.3 bccconfig.py

This is the bbc.py saved configuration. It is written to during program operation and when the program exits. It is read into bcc.py when the program starts. It is suggested not to manually edit this file except to add your cell phone number to allow SMS Alarm texting to work. Please only put your cell phone number in the config file so as not to send unwanted texts to random persons.

4.3.4 bcc.desktop

This is a desktop icon that should be copied to /home/bcc/Desktop. This will allow you to open bcc.py in a Graphical User Interface. It sizes the terminal window so as to display the output of bcc.py properly.

4.3.5 su-bcc.run

This is a script file to automatically run bcc.py as root. You can start bcc.py this way in a terminal window. You may have to size the terminal window to properly display bcc.py output and refresh the screen with the F – Refresh menu option. Using the bcc.destop icon is the preferred method to start bcc.py.

4.3.6 .gp and .dat files

These are files used by gnuplot to show a graph of a brew session.

4.4 Charts

The charting function in bcc.py uses a separate program called gnuplot. To use the charting function properly you must install gnuplot and gnuplot-x11 (see installation instructions paragraph **3.2**.).

If charting is turned on (**menu option G – Graphics**) data to be charted is written to a data file and a script file is created that is used by gnuplot to create and display a chart of the data.

The script file and data file are named after your brew session information in the format 'Brew Name-Batch Number-Batch Size-Beer Style-Brew Method'

Once there are at least two points of data to chart a chart should display showing the data in a chart. On a new brew session this may take twice the charting interval timing.

The chart now updates automatically and uses the charting interval. Unfortunately by doing this you cannot zoom or scroll the chart area.

To display a chart of a previous brew session data:

- ✓ Open a separate terminal window in VNC
- ✓ cd /usr/local/bcc
- ✓ gnuplot './data/file name of your brew session.gp'