

Preliminary Design Review

525.743

David Sorber

February 11, 2013

Project Concept

- Brewing control system
 - Different types of beer require different fermentation temperatures:
 - Ales: 68°F to 72°F
 - Lagers: 45°F to 55°F
 - Active fermentation can raise beer temperature 10-15 degrees F
 - Use a mini refrigerator to control temperature of beer during fermentation
 - Develop control system to monitor the temperature of the beer and control power to the mini refrigerator thereby regulating the temperature

System Capabilities

- Monitor temperature
 - 2 sensors inside the fermentor
 - 2 sensors inside the refrigerator outside the fermentor
- Enable/disable refrigerator
- Provide a method for locally monitoring system status
 - Walk up to refrigerator and see what's going on
- Provide a method for remotely monitoring system status
 - Sit at my desk at work and see what's going on

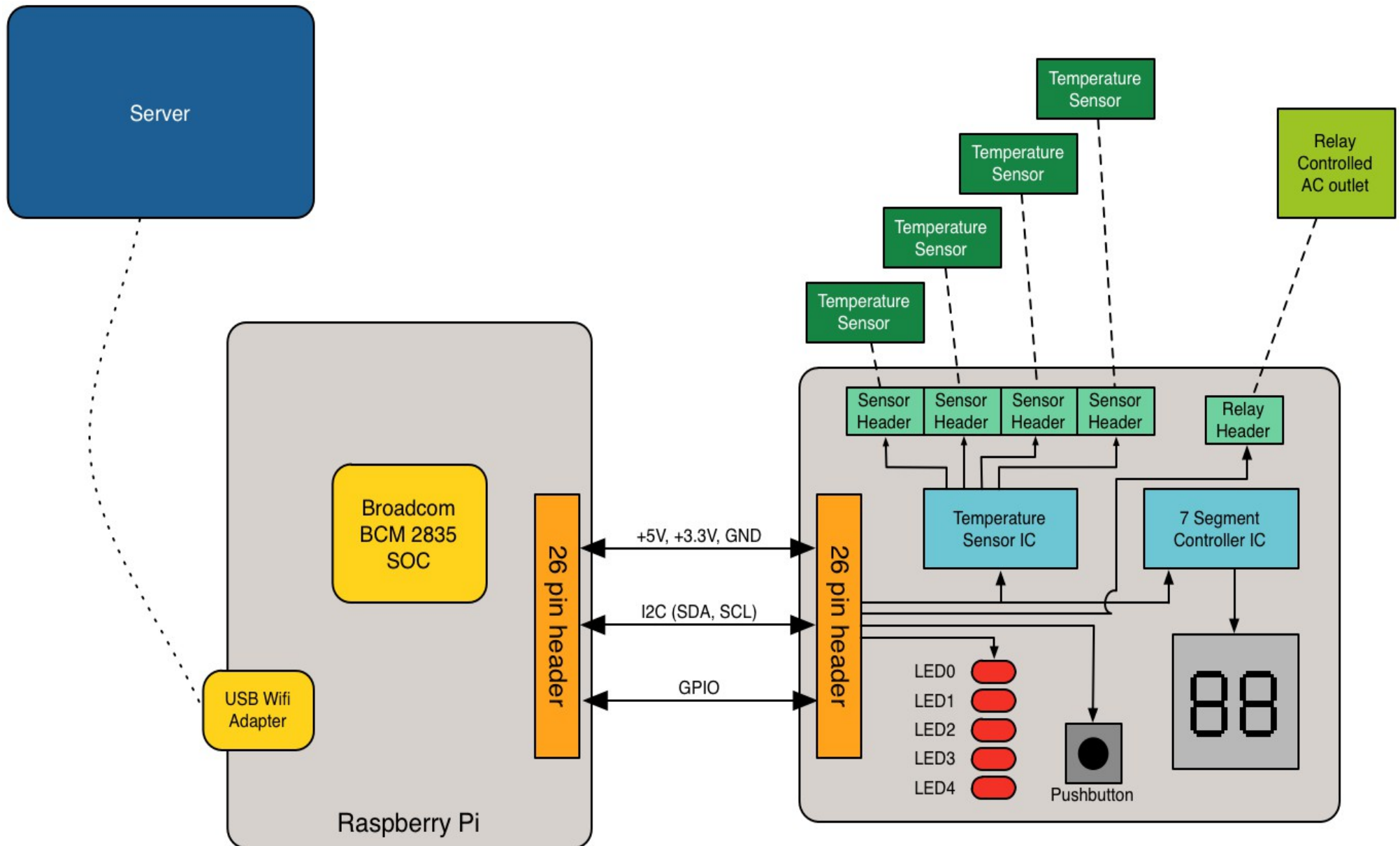
Approach

- Use Raspberry Pi development board
- Develop a daughtercard for the Raspberry Pi
 - Temperature sensor IC
 - Connectors to remote temperature probes
 - 2 digit 7-segment LED
 - 7-segment LED controller
 - Push button to select which temperature sensor is displayed
 - 4 LEDs to indicate which sensor is being displayed
 - 1 LED to indicate if the refrigerator is active
 - Connector to relay controlled AC power outlet
- Use relay controlled AC power outlet to control power to the refrigerator
- Monitor temperature data and send to remote server

References

- <http://www.raspberrypi.org/>
- http://elinux.org/RPi_Low-level_peripherals
- <http://www.raspbian.org/>
- <http://www.ti.com/product/Im95234>
- http://www.nxp.com/products/interface_and_connectivity/i2c/i2c_led_display_control/SAA1064T.html
- http://www.lumex.com/en/products/detail/leds7/led_displays2/thru-hole2/numeric5/dual_dig7/standard30/5610/rounded_segments4/

Block Diagram



Interfaces

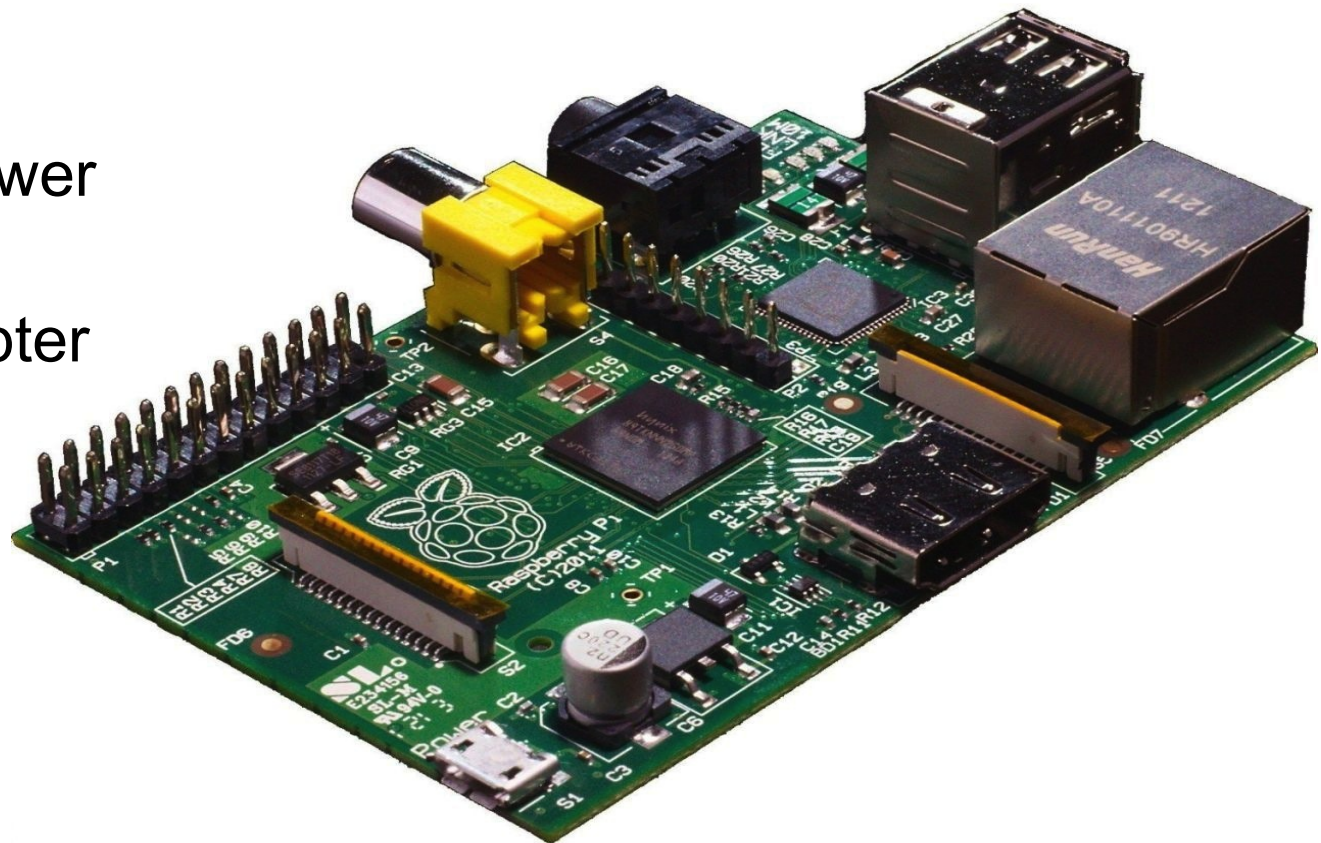
- Raspberry Pi to Server
 - custom protocol on top of TCP/IP via Wifi
- Raspberry Pi to daughtercard
 - 26 pin header (includes +5V, +3.3V, GND, I2C, and GPIO)
- Raspberry Pi to Temperature Sensor IC and 7 segment controller IC
 - I2C
- Raspberry Pi to LEDs, pushbutton, and relay header
 - GPIO
- Temperature sensor header to temperature sensor
 - physical wire
- Relay header to relay
 - physical wire

Software

- Client side
 - Raspberry Pi will run Raspbian Linux
 - Version of Debian Linux created to support the Raspberry Pi
 - Includes Python modules for accessing I2C and GPIO
 - Develop daemon in Python to read temperature sensors, send output to server, control relay, respond to pushbutton, light up appropriate LEDs
- Server side
 - Server runs Ubuntu Linux 12.04 Server
 - Write Python server app to receive and store daemon output data
 - Write WSGI web app to display status

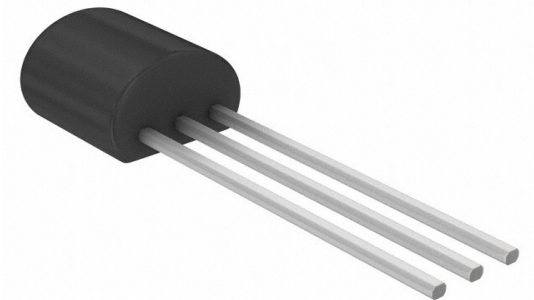
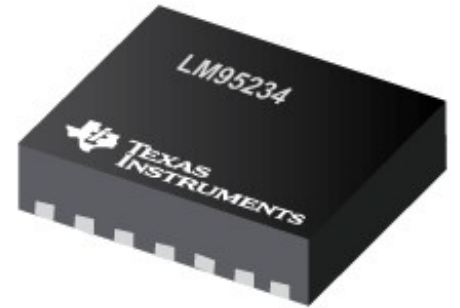
Materials

- Raspberry Pi
 - Status: already own
 - Peripherals:
 - SD card
 - Micro USB power adapter
 - USB Wifi adapter



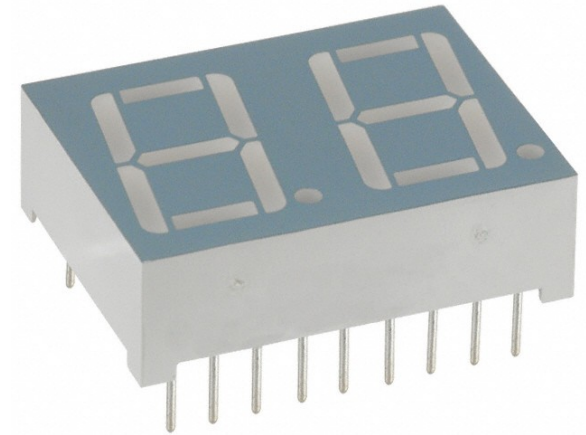
Materials continued

- Daughtercard
 - Status: will design and have fabricated
- Temperature Sensor IC
 - TI LM95234 Quad Remote Diode and Local Temperature Sensor w/ SMBus Interface
 - Status: samples ordered from TI
- Remote Temperature Sensors
 - 3904 NPN BJT Transistors
 - Status: will purchase



Materials continued

- 7 Segment Display
 - Lumex Opto LDD-A516RI
 - Status: samples requested;
will purchase if necessary
- 7 Seg Display Controller IC
 - NXP SAA1064T LED Driver w/ I2C Bus
 - Status: will attempt to get samples;
otherwise will purchase
- Relay controlled AC Outlet
 - PowerSwitch Tail II
 - Status: will purchase



Materials continued

- LEDs, pushbutton switch, miscellaneous passive components, headers, wire, etc.
 - Status: will use whatever I already have, purchase the rest as needed
- Test equipment, solder station
 - Status: will use resources at work

Development Plan

- Design daughtercard and have it fabricated; estimated turnaround time for PCB fab is ~2 weeks (design 2 weeks, fab 2 weeks)
- Develop server application and sample client daemon for testing (2 weeks, occurs during PCB fab delay)
- Assemble daughtercard and perform electrical testing (1 week)
- Develop test programs, in Python, for daughtercard: (2 weeks)
 - Read and debounce pushbutton
 - Activate relay GPIO
 - Turn on and off all standalone LEDs
 - Display digits on the 7 segment display
 - Read temperature values from all four sensors
- Integrate all test programs and sample client daemon into final client daemon (1 week)
- Assemble and test entire system (1 week)
- Brew some beer!

Risk

- I have never designed a PCB before
 - Mitigation: have very experienced coworker to go to for CAD tool introduction and questions
- I have done very little surface mount soldering
 - Mitigation: bribe one of the technicians at work to give me some pointers; practice using junk PCBs at work
- Component availability
 - Mitigation: use several distributors if necessary