Beer Fermentation Control System - Critical Design Review

David Sorber
525.743
Februrary 25, 2013

Project Description

The purpose of this project is to develop a control system to control the temperature of a batch of home-brewed beer while it is fermenting. Different types of beer require different fermentation temperatures. Ales typically ferment between 68°F to 72°F, while lagers typically ferment between 45°F to 55°F. This control system will allow either type of beer to ferment at the optimum temperature regardless of the ambient temperature.

After the beer has been transferred into a fermenter, the fermenter will be placed into a mini refrigerator which will be used to cool the beer while it is fermenting. The control system will then monitor the temperature of the beer inside the fermenter as well as outside the fermenter (but inside the refrigerator). The control system will also provide both local and remote system monitoring capabilities.

Capabilities

- monitor the temperature of beer while it is fermenting
- enable power to the refrigerator when the beer temperature exceeds a predefined threshold temperature
- disable power to the refrigerator when the beer temperature falls below a predefined threshold temperature
- provide a means to locally monitor system status
- provide a means to remotely monitor system status

Limitations

This project is not intended to interface directly with the control circuitry of the mini refrigerator. Instead, the refrigerator will be set on its highest (i.e. coldest) setting and power to the refrigerator will be turned on and off as needed in order to regulate the internal temperature.

Approach

The control system will use a Raspberry Pi computer. The Raspberry Pi is a small, low cost, ARM based computer which is powerful enough to run a standard desktop version of the Linux operating system. There is a large hobbyist community that actively provides support for all things related to the Raspberry Pi. The Raspberry Pi includes an ARM CPU, 512MB of RAM, 2 USB ports, 10/100 Ethernet, HDMI video output, component video output, analog audio output, and a 26 pin header that includes power, ground, I^2C, SPI, and several GPIOs. An IEEE 802.11 wireless (i.e. “WIFI”) adapter will be added to the Raspberry Pi to allow remote communication for monitoring purposes. The Raspberry Pi will periodically send temperature and relay status information to a remote web server which will allow the status of of the entire control system to be monitored remotely.

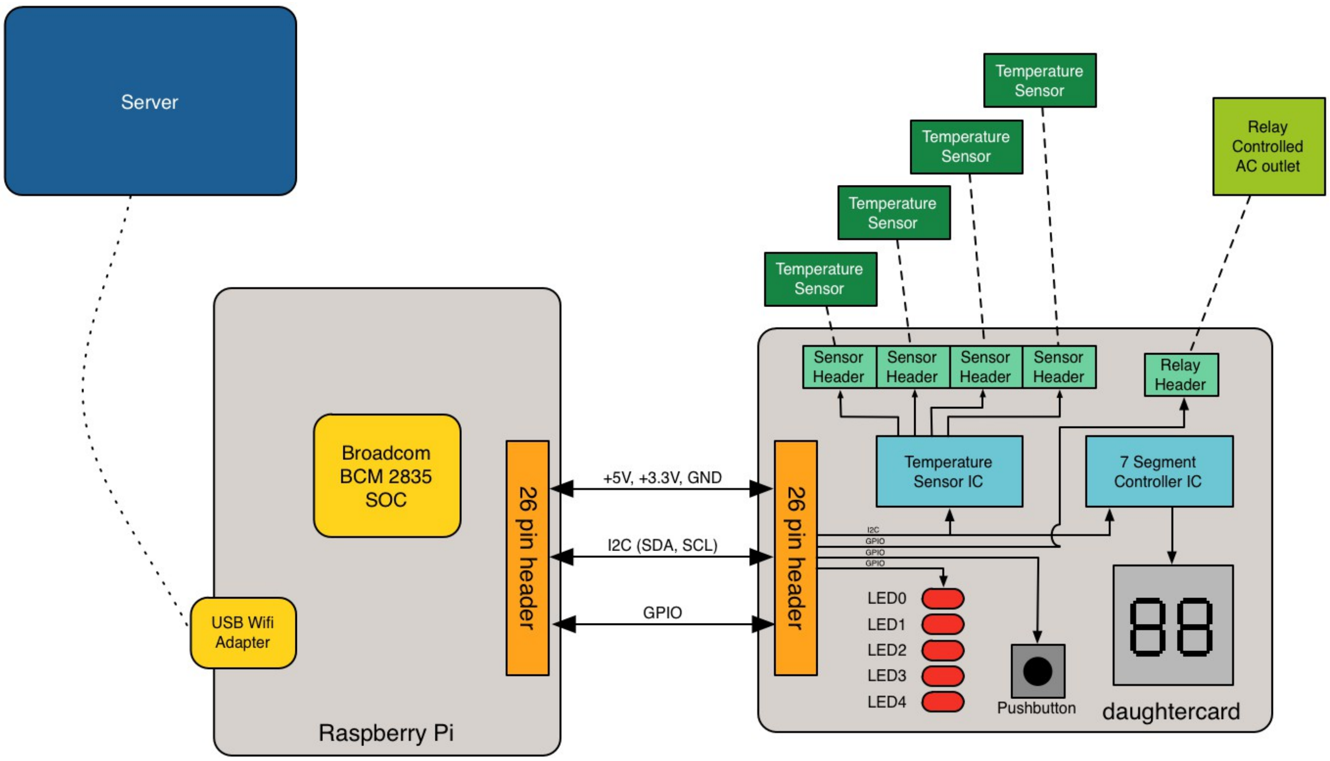
A “daughtercard” will be developed to provide functionality that is specific to the control system but is not found on the Raspberry Pi. The daughtercard will be attached to the Raspberry Pi using the 26 pin header mentioned above. The daughtercard will contain components necessary for remotely sensing temperature and for controlling the external relay to control the mini refridgerator. The daughtercard will also contain several LEDs and a pushbutton to allow the temperature of each remote sensor and the status of the external relay to be monitored locally.

Four remote temperature sensors will be used. Two will be placed inside the fermentor itself to monitor the temperature of the beer more directly. Two additional temperature sensors will be placed inside the refrigerator but outside of the fermentor to monitor the ambient temperature inside the refrigerator. This sensor arrangement will allow both temperature regions to be monitored and correlated.

An external DC relay will be used to control the 120 volt AC (i.e. “wall power”) to the mini refrigerator. The relay will be connected to a header on the daughtercard and controlled by a DC voltage.

Functional Description

Block Diagram

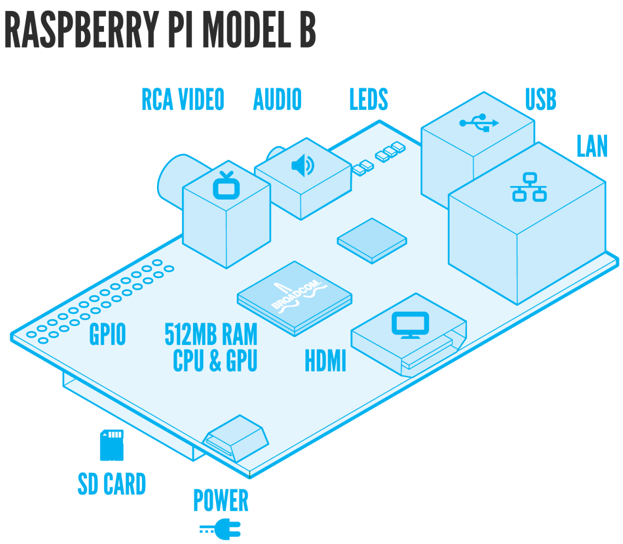


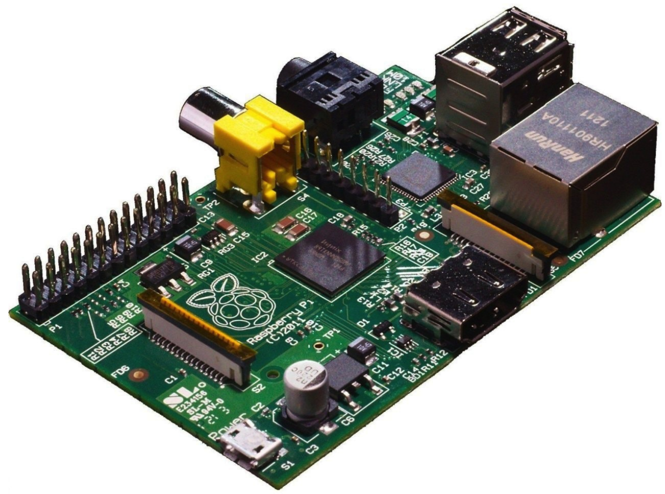
Functional Component Breakdown

- Raspberry Pi board
  - Broadcom BCM2835 SOC
    - ARM1176JZ-F CPU
    - 512 MB RAM
  - USB Wifi adapter
- daughtercard
  - 26 pin (2x13) header
  - remote temperature sensor headers
  - temperature sensor controller
  - 7 segment display
  - 7 segment display controller
  - relay header
  - pushbutton
  - LEDs
- DC controlled 120V AC relay
- remote web server

Materials

Raspberry Pi



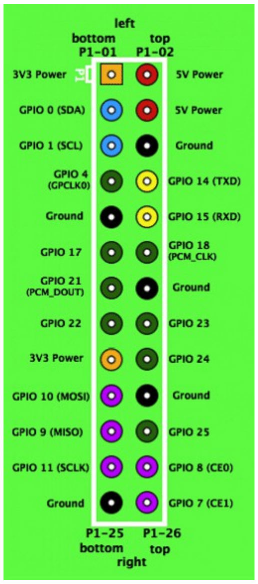


The Raspberry Pi is a small, low cost, ARM based computer which is powerful enough to run a standard desktop version of the Linux operating system. There is a large hobbyist community that actively provides support for all things related to the Raspberry Pi. The Raspberry Pi includes an ARM CPU, 512MB of RAM, 2 USB ports, 10/100 Ethernet, HDMI video output, component video output, analog audio output, and a 26 pin header.

The Raspberry Pi will run the Raspbian Linux distribution which is based off of the Debian Linux distribution and is customized to fully support the Raspberry Pi platform. Raspbian includes a large repository of precompiled software packages for the Raspberry Pi.

The Raspberry Pi will use the following peripherals:

- micro USB power adapter – MCM MWS5-0501000UC/M - External plugin 1A micro USB
  - The power adapter will supply power to the Raspberry Pi and all attached peripherals including the daughter card.
- SD card – SanDisk Extreme 16 GB SDHC Class 10 UHS-1 Flash Memory Card SDSDX-016G-X46
  - The SD card will be the primary nonvolatile storage for the Raspberry Pi and will hold the Raspbian Linux operating system along with all code needed by the system. A 2 GB or larger SD card is required by Raspbian, so this 16 GB card will have ample free space.
- USB Wifi Adapter – Edimax EW-7811Un 150 Mbps Wireless 11n Nano Size USB Adapter
  - The wifi adapter will allow the Raspberry Pi to connect to an existing Wifi network and send data to a web server on the network. The Edimax EW-7811Un contains a Realtek RTL8188CUS chip. This chipset is supported by the open source rtl8192cu driver, which is included by default with Raspbian Linux. The EW-7811Un is powered entirely via the on board USB ports. The EW-7811Un has been confirmed to work both by the Raspberry Pi community and by my own testing.



The Raspberry Pi includes a 26 pin (2 x 13 pin) header. The header includes connections for +5 V, +3.3 V, ground, and several GPIOs which operate at 3.3 V and can be configured to source or sink 2 mA to 16 mA. Several of the GPIOs are dual use can be used for I^2C, SPI, PWM, and a serial UART. Each GPIO can be configured to interrupt on a high level, low level, rising edge, falling edge, or any change.

For the power pins on the header, the maximum current draw is 50 mA for the 3.3 V pins and 300 mA for the 5 V pins.

Daughtercard

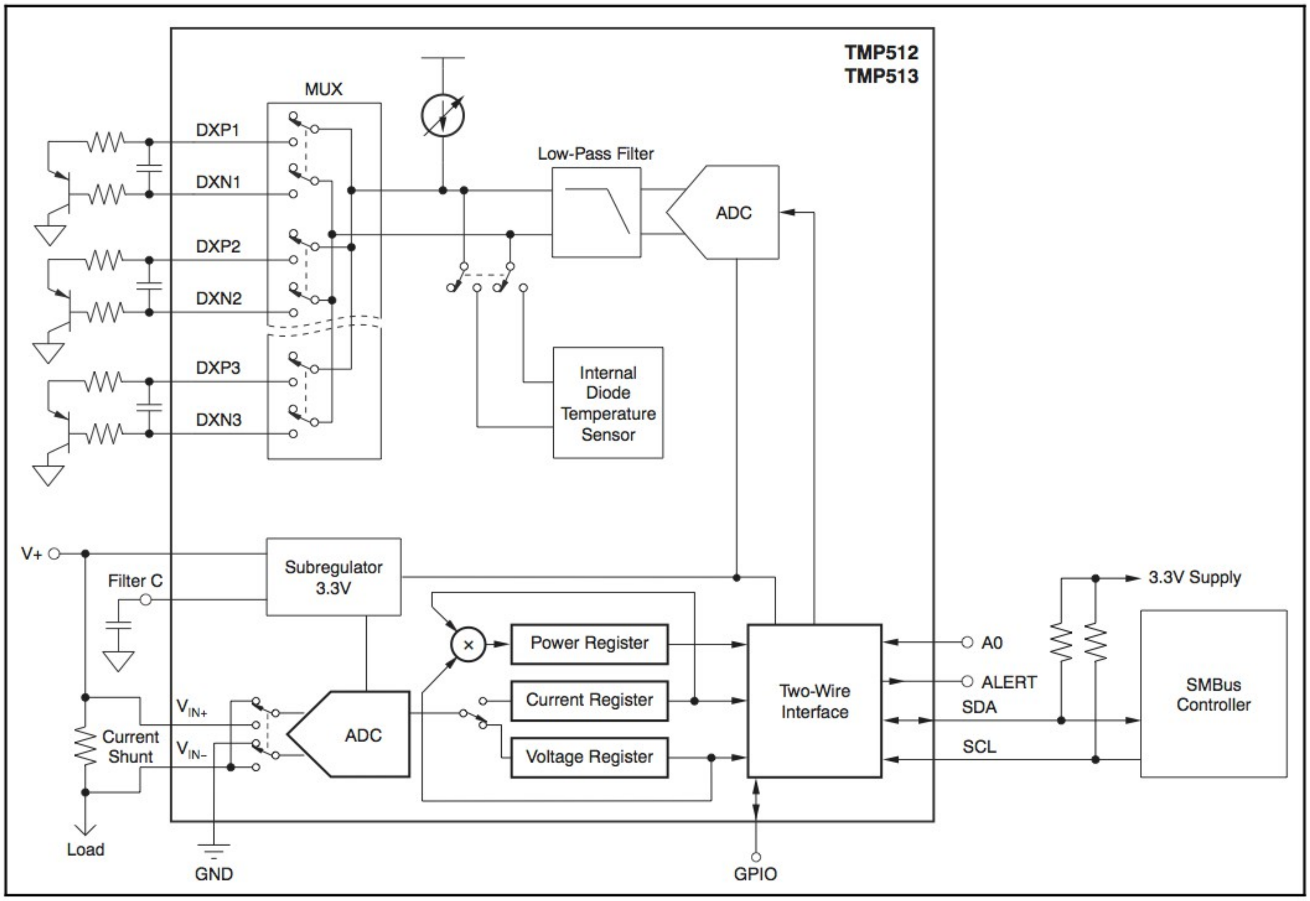
The daughtercard will be a custom PCB that I will design, assemble and test. The PCB will be designed using Altium Designer 10. I have access to this software at my workplace. I also have access to several coworkers who are very experienced users of Altium Designer. My workplace will also provide the necessary soldering and test equipment.

The daughtercard will be fabricated by OSH Park. OSH Park charges $5 per square inch for 3 copies of a 2 layer design. They advertise a 12 day typical turnaround time. OSH Park also fabricates 4 layer designs and charges $10 per inch for 3 copies.

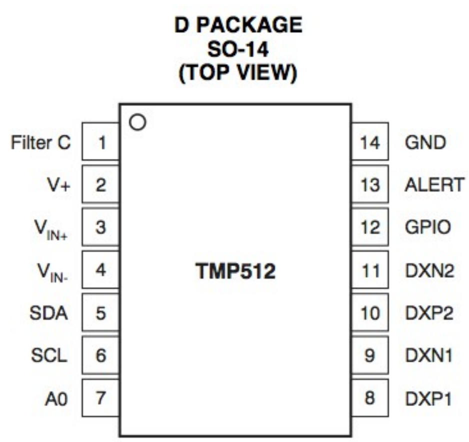
Temperature Sensor Controller

NOTE: I proposed using theTI LM95234 in my PDR however the LM95234 is only available in a QFN package that would be nearly impossible to hand solder. Therefore I selected a different part that is available in a package that is easier to hand solder.

The Texas Instruments TMP512 will be used to control the remote temperature sensors. The TMP512 provides two remote temperature sensor channels, a local temperature sensor, and a current shunt monitor. The local temperature sensor and the current shunt monitor will not be used for the purposes of this project. The TMP512 uses remote diode-connected NPN or PNP type transistors to sense temperature. The TMP512 uses a 2 wire SMBus interface (SCL and SDA) for configuration and data. SMBus is compatible with the I^2C bus provided by the Raspberry Pi (see more details in the I^2C interface section). The TMP512 can be powered from a single +3V to +26V supply and draws a maximum of 1.4mA of current.



In order to support four remote temperature sensors, two TMP512's will be used. In order to connect both TMP512's to the same SMBus/I^{2}C bus, each device must be configured to have a separate slave address. The A0 pin was designed to provide this functionality. When the A0 pin is connected to ground the slave address of the device is set to 0x7C. When the A0 pin is connected to V_{DD} the slave address of the device is set to 0X7D. Appropriately setting the A0 pin will allow two TMP512's to be connected to the same bus.

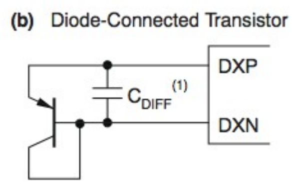


The TMP512 comes in two packages, a 14 pin SOIC package and a 16 pin QFN package. The 14 pin SOIC package will be used to facilitate hand soldering.

Remote Temperature Sensors

The remote temperature sensors will be standard discrete PNP transistors. The Fairchild 2N3906 in the TO-92 through-hole package has been selected.

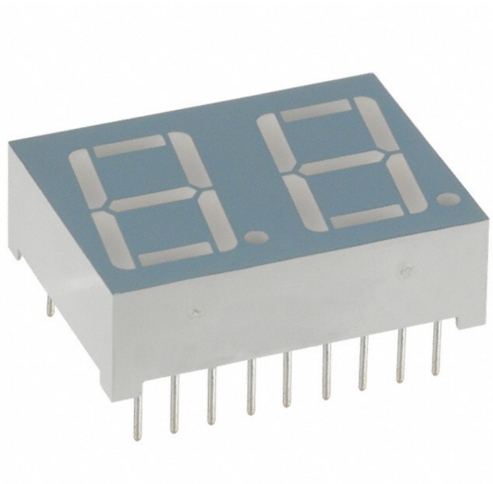
The 2N3906 will be diode-connected such that the collector is connected to the base and the base emitter junction is used for temperature sensing.

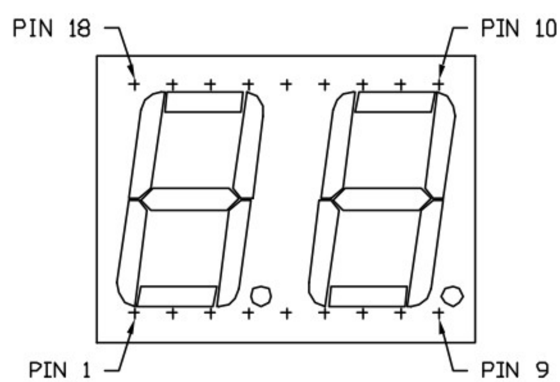


The TMP512 datasheet recommends adding a bypass (i.e. filtering) capacitor between the DXP and DXN inputs to minimize any noise effects. The recommended value of the filtering capacitor is 100pF to 1nF, therefore a 560pF capacitor will be used.

The two remote temperature sensors that will be placed into the fermentor (and therefore in direct contact with the beer) will be sealed in food-grade plastic tubing and sanitized before placement.

7 Segment Display

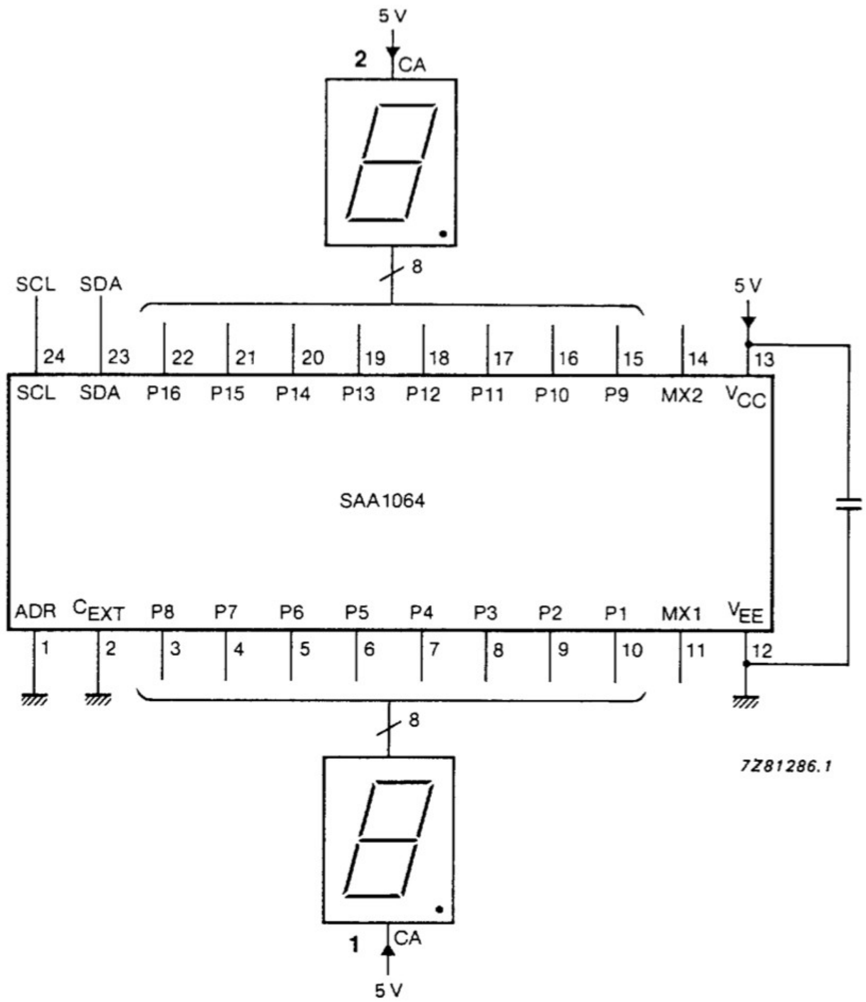




A 7 segment display will be used to display the temperature readings when requested by the user. The Lumex Optoelectronics LDD-A516RI two digit 7 segment display will be used. The LDD-A516RI is a red common anode LED display that includes two digits and two decimal points for a total of 18 pins (including the two anodes). The LDD-A516RI requires a max input current of 30 mA, with a typical input current of 10 mA, and has a forward voltage of 1.7 volts.

7 Segment Display Controller

In order to control the 7 segment display from the I^2C bus a separate driver/controller chip is required. The NXP SAA1064 has been selected for this purpose. The SAA1064 is an I^2C bus interface LED driver that is capable of driving up to four 7 segment digits. The SAA1064 requires an input voltage of 5V and features configurable current sink outputs that can sink up to 21 mA.



The SAA1064 is available in a 24 pin SOIC package. The SAA1064 features an address pin that allows multiple SAA1064's to be used on the same I^2C bus. If the ADR pin is grounded then the I^2C slave address is set to 0x70, which will be sufficient for the one SAA1064 in this design.

PINNING

SYMBOL	PIN	DESCRIPTION
ADR	1	I^2C-Bus slave address input
C_{EXT}	2	external control
P8 to P1	3-10	segment output
MX1	11	multiplex output
V_{EE}	12	ground
V_{CC}	13	positive supply
MX2	14	multiplex output
P9 to P16	15-22	segment output
SDA	23	I^2C-Bus serial data line
SCL	24	I^2C-Bus serial clock line

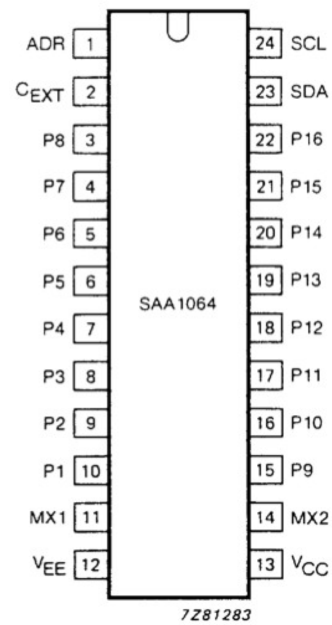
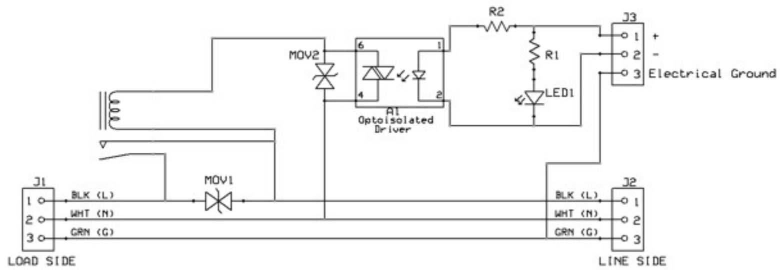


Fig.2 Pinning diagram.

DC controlled 120V AC relay

The PowerSwitch Tail II is a DC voltage controlled 120 VAC relay. The PowerSwitch Tail II is intended to allow devices powered by 120 VAC wall power to be controlled by small DC voltages from a microcontroller development board.

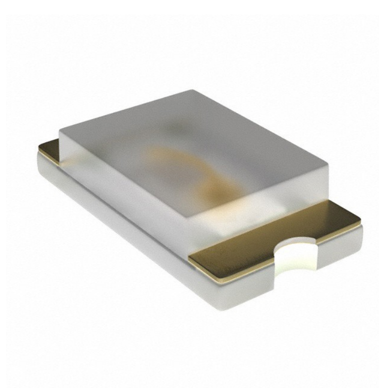


The PowerSwitch Tail II switches up to 15 A ,120 VAC with a 3 mA, 3 VDC to 30 mA, 12 VDC control signal.

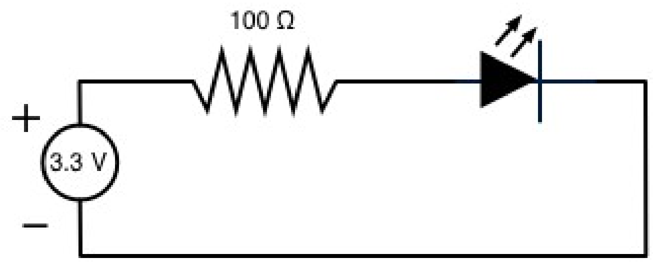
<u>Pushbutton</u>

The Tyco Electronics Connectivity FSM2JH off-momentary pushbutton switch will be connected to a GPIO input on the Raspberry Pi. It will used to provide input to the Raspberry Pi and cycle through each of the temperature sensors.

<u>LEDs</u>



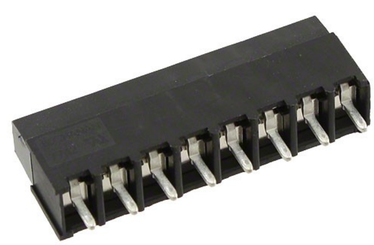
Five discrete LEDs will be used on the daughtercard. Four will be used to indicate which temperature sensor's value is currently being displayed on the 7 segment display. The fifth LED will be used to indicate if the external AC relay is on or off. The OSRAM Optoelectronics LB Q39G-L2N2-35-1 blue 0603 package LED will be used.



The LB Q39G-L2N2-35-1 requires a typical operating current of 5 mA and has a 2.85 V forward voltage. The LEDs will be connected to 3.3 V GPIO outputs from the Raspberry Pi. Therefore each LED will need a 100 Ω (rounded up from 90 Ω) current limiting resistor in series.

<u>Headers</u>

The external AC relay will be connected to the daughtercard via wires attached to a two conductor screw terminal header. The Tyco Electronics Connectivity 282836-2 has been selected.



The four remote temperature sensors, each with two leads, will be connected via wires attached to an 8 conductor screw terminal header. The Tyco Electronics Connectivity 796949-8 has been selected.

Interface Descriptions

Raspberry Pi to Web Server

The Raspberry Pi will be connected to the web server via Wifi. The Raspberry Pi will communicate with the web server using a custom protocol on top of TCP/IP. A daemon (written in Python) running on the Raspberry Pi will open a TCP socket connection to a daemon (also written in Python) on the web server. The daemon running on the Raspberry Pi will then send serialized (pickled in Python-speak) Python objects to the daemon on the web server. The daemon on the web server will receive each serialized object, deserialize (unpickle) it and then process it according to its type. The following types of objects containing the following instance data will be used:

- temperature reading object
  - sensor – indicates which sensor the temperature reading came from
  - timestamp – indicates when the temperature value was measured
  - temperature – the actual temperature value recorded by the sensor
- relay status object
  - relay status – indicates if the relay was turned on or off
  - timestamp – indicates when the relay was turned on or off

The daemon running on the Raspberry Pi will send temperature readings from all four sensors to the daemon running on the web server once every minute. The daemon running on the web server will record the temperature data as it is received. The daemon running on the Raspberry Pi will send the relay status whenever the relay is switched on or off.

Raspberry Pi to Daughtercard

A 26 conductor ribbon cable will be used to physically connect the header on the Raspberry Pi to the header on the daughtercard. This connection includes 5 V, 3.3 V, ground, I^2C bus, and several GPIOs as mentioned previously.

SMBus/I^2C Bus

The SMBus used by the TMP512 was derived from the I^2C bus specification, however the SMBus voltage and timing requirements are more strictly defined than those for the I^2C bus. As long as the SMBus specifications are met it is entirely possible for SMBus and I^2C bus devices to coexist and interoperate on the same bus.

Since SMBus is a subset of I^2C bus specification, a SMBus will be compatible with I^2C bus capable devices. Therefore the two wire bus implemented on the daughtercard and controlled by the Raspberry Pi will conform with the SMBus specification. The electrical and important timing characteristics of the SMBus specification are as follows:

- V_{DD} – min 2.7 V, max 5.5 V
- V_{IL} (input low voltage) – max 0.8 V
- V_{IH} (input high voltage) – min 2.1 V, max V_{DD}
- I_{PULLUP} (pull-up resistor current) – min 100 μA, max 350 μA
- F_{SMB} (SMBus frequency) – min 10 KHz, max 100 KHz

For the purposes of this project, V_{DD} will be tied to 3.3 V so a pull up resistor of at least 9.5 kΩ is required order to not exceed the maximum current value of 350 μA specified by the SMBus specification.

Temperature Sensor Header to Temperature Sensor

The temperature sensor transistors will be physically connected via wire to the headers on the daughtercard.

Relay Header to Relay

The relay will be physically connection via wire to the header on the daughtercard.

Software

A daemon will be developed, in Python, to run on the Raspberry Pi. This daemon will have the following functionality:

- Read the value of each temperature sensor once every minute
- Send temperature readings to the web server daemon once every minute
- If the temperature exceeds a predefined threshold value enable the relay (turn on the refrigerator)
- If the temperature falls below a predefined threshold value disable the relay (turn off the refrigerator)
- Send relay status messages to the web server when the status of the relay changes
- Read the pushbutton input looking for presses
    - Each time the pushbutton is pressed display the last read temperature value for the next temperature sensor

A separate daemon will be developed, in Python, to run on the web server. This daemon will receive messages from the Raspberry Pi, compare the received values to the previous values and insert into a database any values that differ.

A very simple WSGI web application will be developed to query the database and display the results in table form in descending order such that the most recent temperature or relay status change appears at the top.

Development Plan

- Design daughtercard using Altium Designer. Then have daughtercard fabricated. Estimated turnaround time for PCB fab is ~2 weeks. (1 week design, 2 weeks fab)
- Develop server daemon and sample Raspberry Pi daemon, while PCB is being fabricated. Also begin developing sample test programs using Raspberry Pi and a breadboard. (2 weeks during PCB fab)
- Assemble daughtercard and perform electrical testing. (1 week)
- Develop test programs to exercise single elements of the daughtercard: (1 week)
    - Read and debounce pushbutton
    - Activate relay using GPIO from Raspberry Pi
    - Illuminate each standalone LED
    - Display digits on the 7 segment display
    - Read temperature values from all four remote sensors
- Integrate all test programs together with the sample Raspberry Pi daemon to create the final version of the software. (1 week)
- Assemble and test the entire system. (1 week)

References

1. SAA1064 4-digit LED-driver with I2C-Bus interface Data Sheet
2. 2N3906 / MMBT3906 / PZT3906 PNP General Purpose Amplifier Data Sheet
3. Broadcom BCM2835 ARM Peripherals
4. LB Q39G Data Sheet
5. LDD-A516RI Rev C Drawing
6. PowerSwitch Tail II Product Insert August 2012
7. Temperature and Power Supply SystemMonitors TMP512/3 - SBOS491A
8. System Management Bus (SMBus) Specification Version 2.0
9. Maxim Integrated Application Note 476: Comparing the I^2C Bus to the SMBus