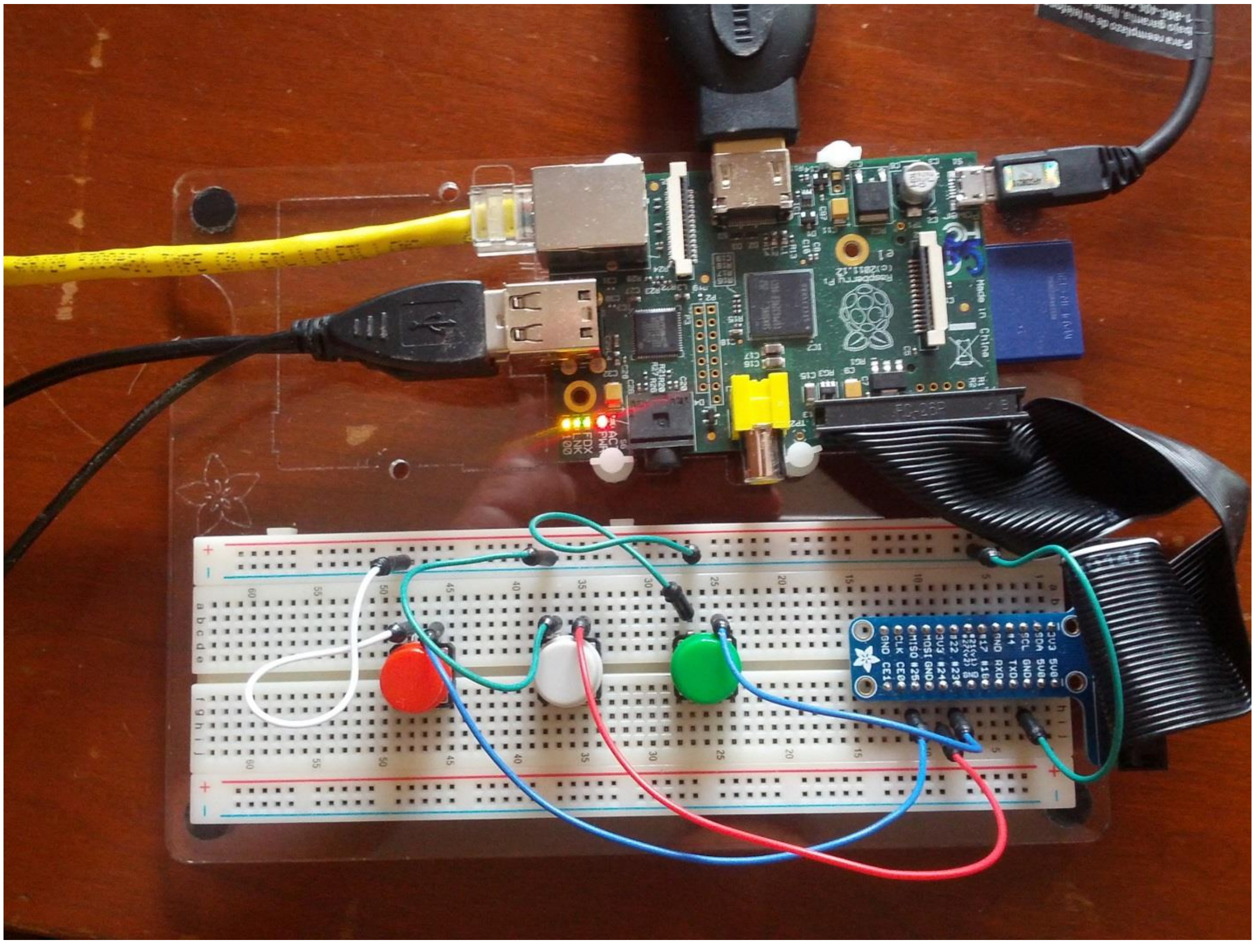
At the 2013 Aras ACE conference, Razorleaf presented a proof-of-concept project using a Raspberry Pi as a special purpose Aras Innovator client. The special purpose client was intended to simulate a shop floor system to deliver assembly instructions. The assembly instructions are managed inside Innovator.

If you are not familiar with the Raspberry Pi is a credit card sized, single board computer that runs Linux. This was my first project using Linux and I decided to use the Python programming language. The project also took advantage of the Raspberry Pi's general purpose IO (GPIO) to provide the user a simple hardware button interface to advance through the assembly instructions. Here is a picture of the pi and the wiring for the buttons.



Being my first experience with Linux and Python, I was unsure what would be required to provide a user interface. I decided to implement it using the Pygame library. This turned out to be much more difficult to develop than the code that communicates with Innovator. I wrote the Python code that communicates with Innovator to take an AML statement and return the result. It's a very simple implementation.

I have posted the Python code and the Innovator package that creates the data model. The general procedure to test this out is to

1. Import the package into a new Aras Innovator solutions database
2. Create a new part
3. Add Assembly Instructions to the part with text and screenshots by going to the part's Assembly Instructions tab.
4. Edit the VaultServerConfig.xml file to add the following tag, <enforce_user_credentials>0</enforce_user_credentials>.
5. Wire the buttons to the Raspberry Pi as shown above
6. Copy the Python code to the Raspberry Pi
7. Edit the araslib.py file with the variable values for your Innovator environment.
8. From the Raspberry Pi, run arasClientUi.py with sudo python arasClientUi.py
9. From the UI that is displayed from the pi, type in the part number of the part you created above and press enter. Note that the code does not take input from the number pad of the keyboard.
10. The first assembly instruction should be presented.
11. Press the green button to advance to the next instruction and the white button to go to the previous instruction. After cycling through the instructions, the client will create a "serial number" and add a record to the part on the Serial Number tab. The serial number will include who built the part, when it was built, and how long it took to build it.
12. While at the part number entry screen, press the red button to exit the program.

If you have any questions, please contact me Dennis.Lindinger@Razorleaf.com