See-3PO: User Guide

A Raspberry Pi Multispectral Imager

1. Introduction

This system is a simple multispectral imager. It runs on a cluster of 5 Raspberry Pi 2 Bs which each has a quad-core and a lot of processing power. The user interface was designed to be as self-explanatory as possible, but this user guide is meant to supplement with information that may not be immediately apparent.

The cameras included in this system are all Raspberry Pi modules. The default cameras and lenses used in this system allow the capture of normal visual light, infrared light (longer wavelength and lower frequency than visual), ultraviolet light (shorter wavelength and higher frequency than visual), and long wave infrared light (longer wavelength and lower frequency than infrared) – also known as thermal.

The unit has around 5 hours of battery life and takes around 3 hours to fully charge. If you wish to remote into the system, a router is required. This increases power consumption and decreases the battery life.

Though the system is ready to use, the advanced use can also adapt it to their particular usages. The lenses covering the cameras can be switched to filter out different wavelengths of the electromagnetic spectrum. Written code has been well documented to allow the user to make edits without disrupting the overall flow of the program. It is recommended that before any changes are made, you document the state of the system with pictures or other notes. Code should be backed up before you make edits so you always have a working set of code.

(as of this writing 2e and 2f are not fully functional)

2. Basic User Guide

To use the imager, first switch it on. The power button is on the back of the unit, directly below the touch screen. Before the imager software begins, the Raspberry Pi runs through start up and you will see a black screen with white text flashing on it. The imager is ready to use when you see the panel in Fig. 1 displayed on the touch screen.

There are six different main systems: the main display window, the main menu, the composite menu, the composite display, the animal eyes display, and the system menu.

For safe shutdown, navigate to Menu then to System Menu and select Shutdown Camera. The system can also be shut down by flicking the power switch below the touch screen.

a. Physical System

The body of the multispectral imager is roughly rectangular. There is a Y shaped piece protruding from the front that houses the cameras and the chosen lenses. The system shown in Fig. 1 does not have any filters on it.

i. Front View

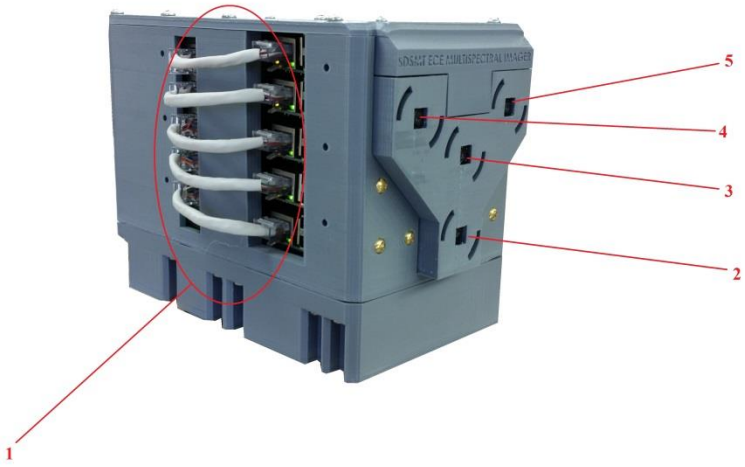

Figure 1. Front/Side view of the system

1. The Ethernet cables connecting the Raspberry Pis and allowing communication between them.

(as of this writing 2e and 2f are not fully functional)

2. The fourth camera in the network. Usually filtered to be the visual camera.
3. The third camera in the network is the long wave infrared or thermal camera. The only camera that is typically unfiltered.
4. The first network camera. Typically the IR camera.
5. The second network camera. Typically the UV camera.

The first system in the network is the Raspberry Pi with the touch screen so the first camera is at network position 2 and the second at network position 3 and so on.

ii. Back View

Figure 2. Back/Side view of the system

1. The power button allows the system to be easily and quickly turned on or off.
2. Ethernet cables are still visible in this view. In your system, they should be covered by a 3D printed shield to protect them from becoming loose.
3. The touch screen where the important information for running the camera and images are displayed.

b. Main Display Window: Multi-Spectral Imager

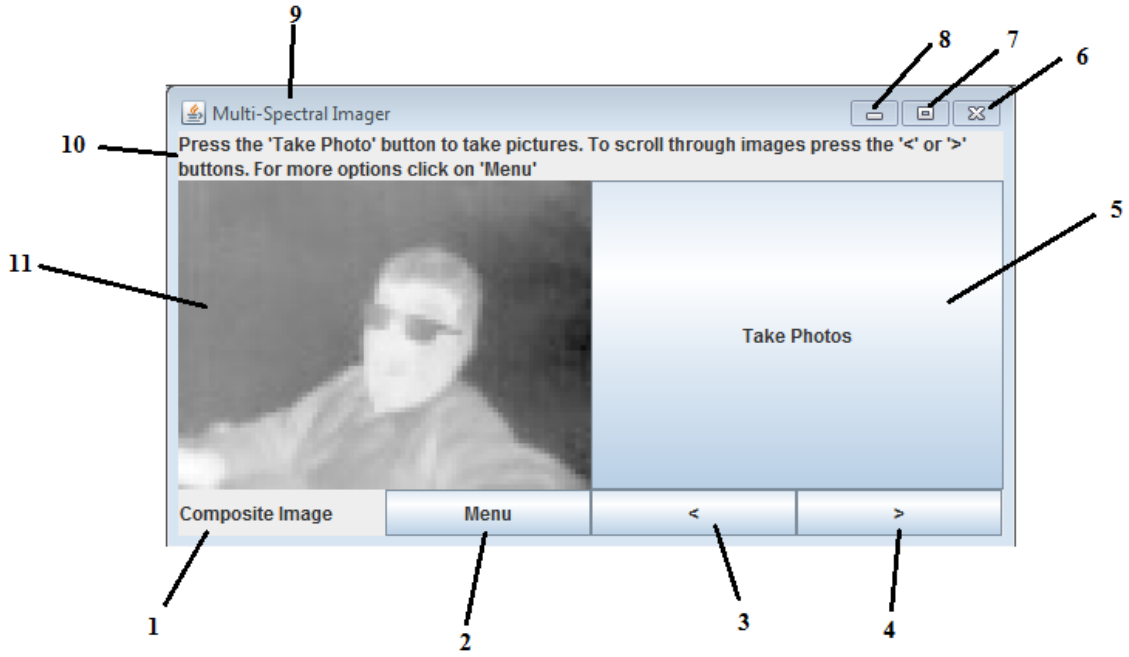

Figure 3. Main display screen

The main display window is used to view and take images. It shows a rotation of images; the default order is: Composite, Visual, Ultraviolet (UV), Infrared (IR), and Long Wave Infrared (Thermal). Fig. 3 above shows an annotated view of the main display screen.

1. Image Label: this label will update as you scroll through the images to tell you explicitly which image you are viewing.
2. Menu Button: this button brings up the menu for additional functionality. (Menu shown in Fig. 4)
3. Previous Button: this button goes back to the previous image that was displayed. 1 & 11 will be updated.
4. Next Button: this button displays the next image. 1 & 11 will be updated.
5. Take Photos Button: this button takes pictures. Buttons 3 & 4 will not work until this button has been pressed.
6. Exit Button: this button will exit the program but not turn off the system. Avoid pressing this button.
7. Maximize Button: this button expands the window to full screen. Avoid pressing this button.
8. Minimize Button: this button will pull down the window without exiting the program. You will not be able to reopen the window easily. Avoid pressing this button.

9. Window Label: this label gives a brief description of what window is currently being displayed.
10. Instruction Label: this label gives a brief description of how to use the displayed window.
11. Image Display: this section is where your images will show up.

c. Main Menu: Menu

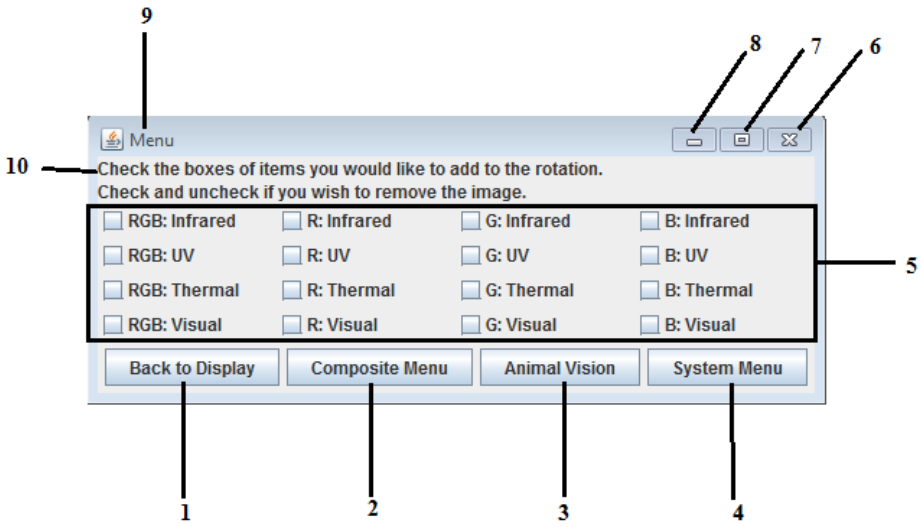

Figure 4. The main menu

The main menu is used to add additional images to the display beyond the default. Clicking a check box adds the image to the rotation. If an image is already in the rotation, then you have to check and uncheck the box to remove the image from the list of images to be displayed.

1. Back to Display Button: this button closes the menu and returns to the main display window.
2. Composite Menu Button: this button opens a menu that allows the user to create their own composite image from the building blocks shown in the checkboxes.
3. Animal Vision Button: this button closes this menu and opens the animal vision menu to let the user see the world the way selected animals do.
4. System Menu Button: this button closes this menu and opens the system menu so the system can be rebooted or shutdown.
5. Check Boxes: these check boxes allow the user to add or remove the images from the rotation on the main display window. Checking a box adds that

image to the rotation. Checking and then unchecking removes that image from the rotation.

6. Exit Button: this button will exit the program but not turn off the system. Avoid pressing this button.
7. Maximize Button: this button expands the window to full screen. Avoid pressing this button.
8. Minimize Button: this button will pull down the window without exiting the program. You will not be able to reopen the window easily. Avoid pressing this button.
9. Window Label: this label gives a brief description of what window is currently being displayed.
10. Instruction Label: this label gives a brief description of how to use the displayed window.

d. Composite Menu: Menu*

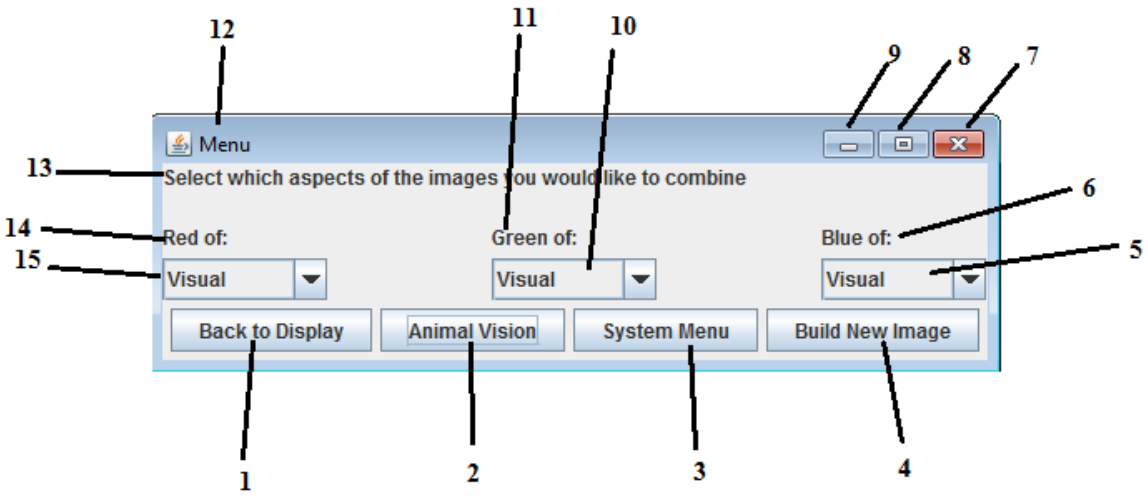

Figure 5. New Composite Image Menu

The composite menu is used to create a new composite image from the user's preferences. The user can select which spectrum they want the red, blue, and green from and then a Python program is called which composites the aspects of the images together.

1. Back to Display Button: this button closes the menu and returns to the main display window.

(as of this writing 2e and 2f are not fully functional)

2. Animal Vision Button: this button closes this menu and opens the animal vision menu to let the user see the world the way selected animals do.
3. System Menu Button: this button closes this menu and opens the system menu so the system can be rebooted or shutdown.
4. Build New Image Button: this button triggers the build new image operation and opens the composite display.
5. Blue Combo Box: this box is where the user selects which image they wish to use the blue color values from to create a new image.
6. Blue Label: this label is used to indicate which color is being selected by the combo box below.
7. Exit Button: this button will exit the program but not turn off the system. Avoid pressing this button.
8. Maximize Button: this button expands the window to full screen. Avoid pressing this button.
9. Minimize Button: this button will pull down the window without exiting the program. You will not be able to reopen the window easily. Avoid pressing this button.
10. Green Combo Box: this box is where the user selects which image they wish to use the green color values from to create a new image.
11. Green Label: this label is used to indicate which color is being selected by the combo box below.
12. Window Label: this label gives a brief description of what window is currently being displayed.
13. Instruction Label: this label gives a brief description of how to use the displayed window.
14. Red Label: this label is used to indicate which color is being selected by the combo box below.
15. Red Combo Box: this box is where the user selects which image they wish to use the red color values from to create a new image.

(as of this writing 2e and 2f are not fully functional)

e. Composite Display: Menu*

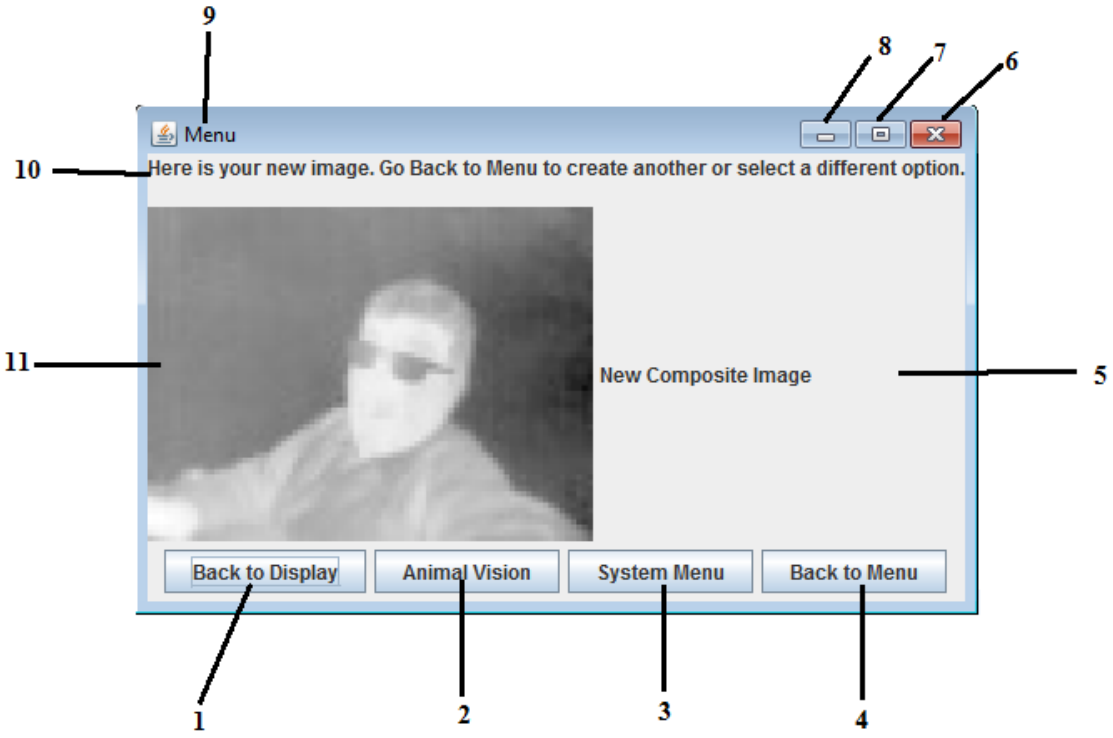

Figure 6. New Composite Image Display

The composite display menu shows the new image created according to the user's preferences. If you want to build another image, you have to go back to menu and select "New Composite Image" to create additional images.

1. Back to Display Button: this button closes the menu and returns to the main display window.
2. Animal Vision Button: this button closes this menu and opens the animal vision menu to let the user see the world the way selected animals do.
3. System Menu Button: this button closes this menu and opens the system menu so the system can be rebooted or shutdown.
4. Back to Menu Button: this button closes this window and returns to the main menu shown in 2c and Fig. 4.
5. Composite Image Label: this label shows the new image that has been created based on the user's selections.
6. Exit Button: this button will exit the program but not turn off the system. Avoid pressing this button.
7. Maximize Button: this button expands the window to full screen. Avoid pressing this button.

(as of this writing 2e and 2f are not fully functional)

8. Minimize Button: this button will pull down the window without exiting the program. You will not be able to reopen the window easily. Avoid pressing this button.
9. Window Label: this label gives a brief description of what window is currently being displayed.
10. Instruction Label: this label gives a brief description of how to use the displayed window.
11. Image Display: this is where the newly created image will be displayed.

f. Animal Vision: Animal Eyes*

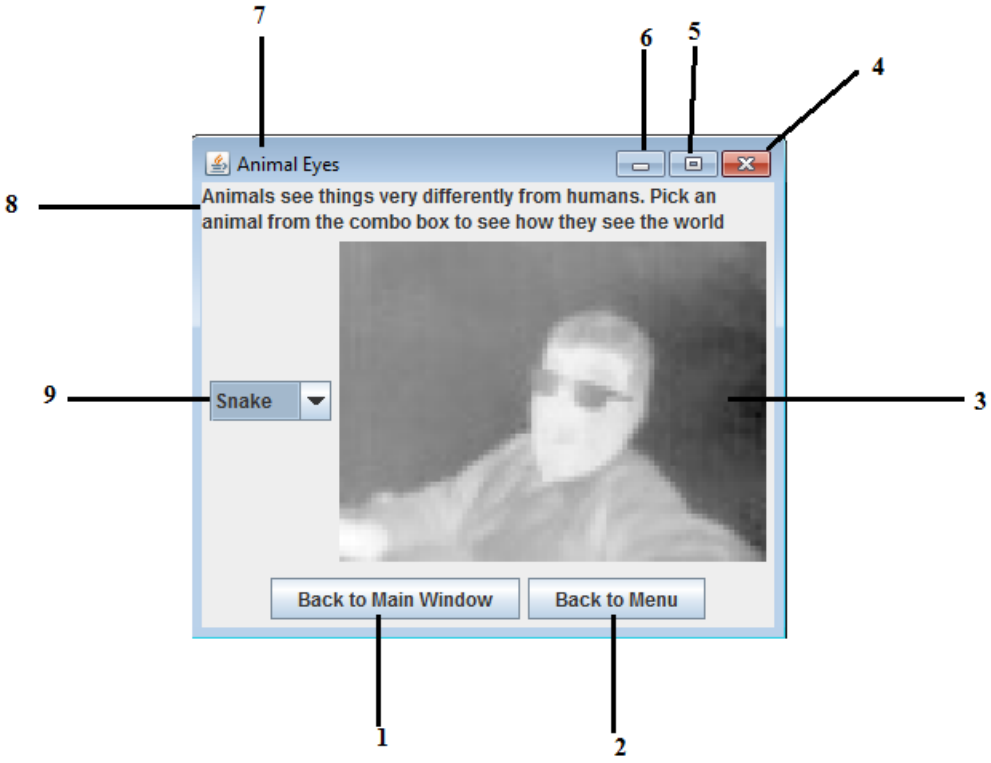

Figure 7. Animal Vision Display

The animal vision display is meant to be a window into the world that animals see. Their perception of the world is different from that of humans because they have different receptors in their eyes that either allow them to see fewer or more bands of the electromagnetic spectrum than us. The animals that were chosen for this system originally are: bee, crayfish, dog, octopus, rabbit, rat, snake, and spider. To switch between the different animal views of the taken image, all you have to do is select another option from the drop down menu.

(as of this writing 2e and 2f are not fully functional)

1. Back to Main Window Button: this button closes the menu and returns to the main display window.
2. Back to Menu Button: this button closes this window and returns to the main menu shown in 2c and Fig. 4.
3. Image Display: this is where the image showing what the animal sees will be displayed.
4. Exit Button: this button will exit the program but not turn off the system. Avoid pressing this button.
5. Maximize Button: this button expands the window to full screen. Avoid pressing this button.
6. Minimize Button: this button will pull down the window without exiting the program. You will not be able to reopen the window easily. Avoid pressing this button.
7. Window Label: this label gives a brief description of what window is currently being displayed.
8. Instruction Label: this label gives a brief description of how to use the displayed window.
9. Animal Combo Box: this combo box holds the names of all the animals we can see through the eyes of. Change the image by selecting another animal from the drop down box.

g. System Menu: System Menu

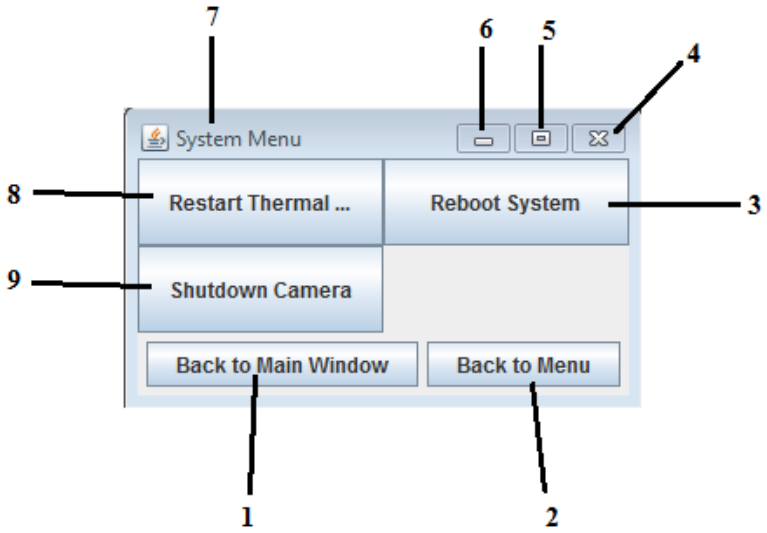

Figure 8. System Menu

(as of this writing 2e and 2f are not fully functional)

The system menu allows the user to control the system's hardware a bit. The three main functions provided here are: restarting the thermal camera, restarting the system, and safely shutting down the software of the system.

1. Back to Main Window Button: this button closes the menu and returns to the main display window.
2. Back to Menu Button: this button closes this window and returns to the main menu shown in 2c and Fig. 4.
3. Reboot System Button: this button runs a python script that restarts the whole system.
4. Exit Button: this button will exit the program but not turn off the system. Avoid pressing this button.
5. Maximize Button: this button expands the window to full screen. Avoid pressing this button.
6. Minimize Button: this button will pull down the window without exiting the program. You will not be able to reopen the window easily. Avoid pressing this button.
7. Window Label: this label gives a brief description of what window is currently being displayed.
8. Restart Thermal Button: this button runs a python program that gives the thermal camera a software reset. This is required every once in a while if the thermal camera is acting up.
9. Shutdown Camera Button: this button runs a python program to shut down the software running on all the Raspberry Pis. It still requires the user to flip the power switch on the back of the unit.

3. Advanced Setup

a. Changing Lenses

Lenses can be changed very easily. 25mm circular filters fit in the modular design of this system. All that is required once you have obtained new filter lenses is to remove the yellow inset on the front of the system, carefully dispense the current lens and store it away safely, put your new filter in, and then pop the yellow inset back into the front of the system.

(as of this writing 2e and 2f are not fully functional)

It is recommended that you edit the code to reflect the change in spectra you are showing, but it is simpler to just remember something like the IR camera is now Blue light in the visible range.

b. Battery Management

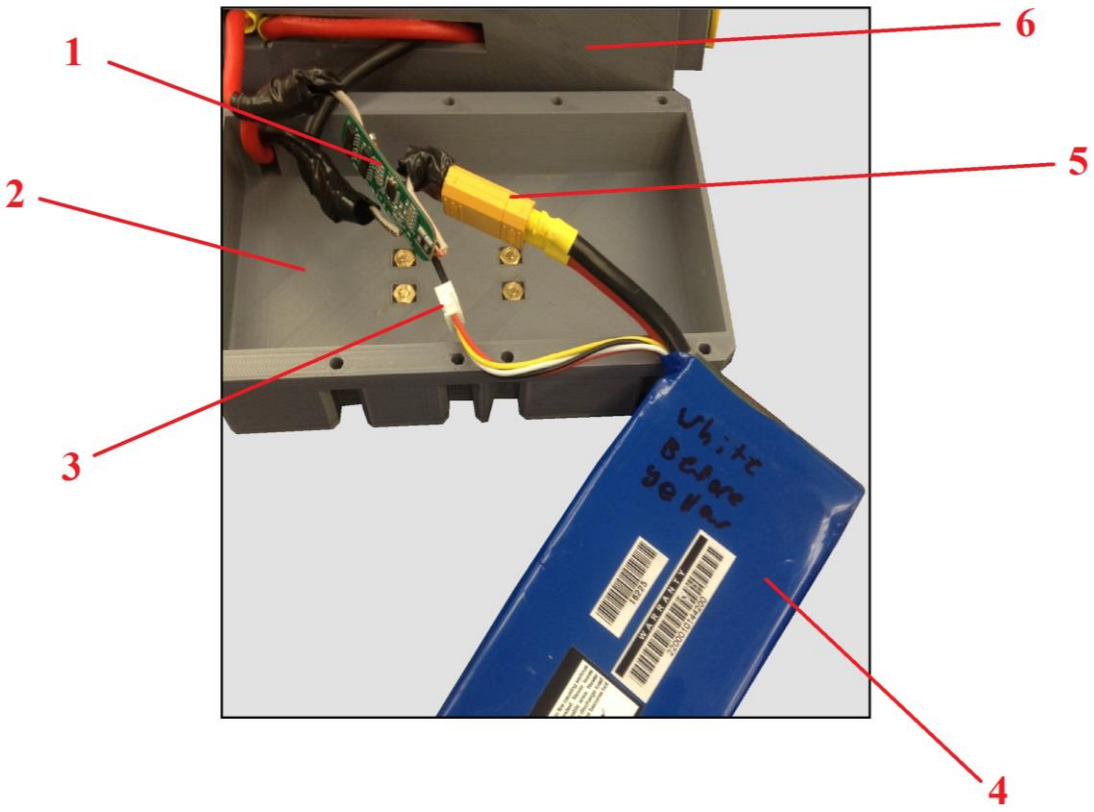

Figure 9. Proper battery attachment

Fig. 9 shows the battery properly attached to the unit. If the battery is removed for charging or replacement, ensure that the white clips, item 3 in the diagram, are connected a minute or so before attaching the yellow clips, item 5 in the diagram.

1. Battery management circuit.
2. Battery case, is screwed to the bottom of the unit.
3. White connectors

(as of this writing 2e and 2f are not fully functional)

4. Battery
5. Yellow connectors
6. Unit

c. Editing Code

The code for the graphical user interface (GUI) was written in Java using Eclipse. It runs on the Raspberry Pis natively and calls Python scripts to communicate with the cameras.

The GUI code was split into three main packages: code.gui, code.handlers, and code.run. The handlers package then had two sub-packages: code.handlers.functions and code.handlers.menu. The purpose of all this division was to keep the code as organized as possible.

i. gui: classes in this package are instantiated to physical representations.
ii. handlers: classes in this package define actions that do not specifically fall under functions or menu.
    1. functions: defines the behaviors for button presses that trigger outside code or programs such as reboot, restart, etc.
    2. menu: defines the button press behaviors that result in the display of a menu or return to a previous menu
iii. run: holds the python script and stubs for programs that are called by the Java code

Organizing the code in this manner kept any one package from becoming overwhelmingly full. If you choose to change this organization scheme, please be sure that you choose refactor and then move to rearrange the classes or packages as this will update any references to them in other classes.

The classes were commented in the JavaDoc standard and the greatest attempt was made to ensure that variable names made sense and were descriptive. TODO comments were left where code was incomplete or there were still some tweaking left to do wherever possible.

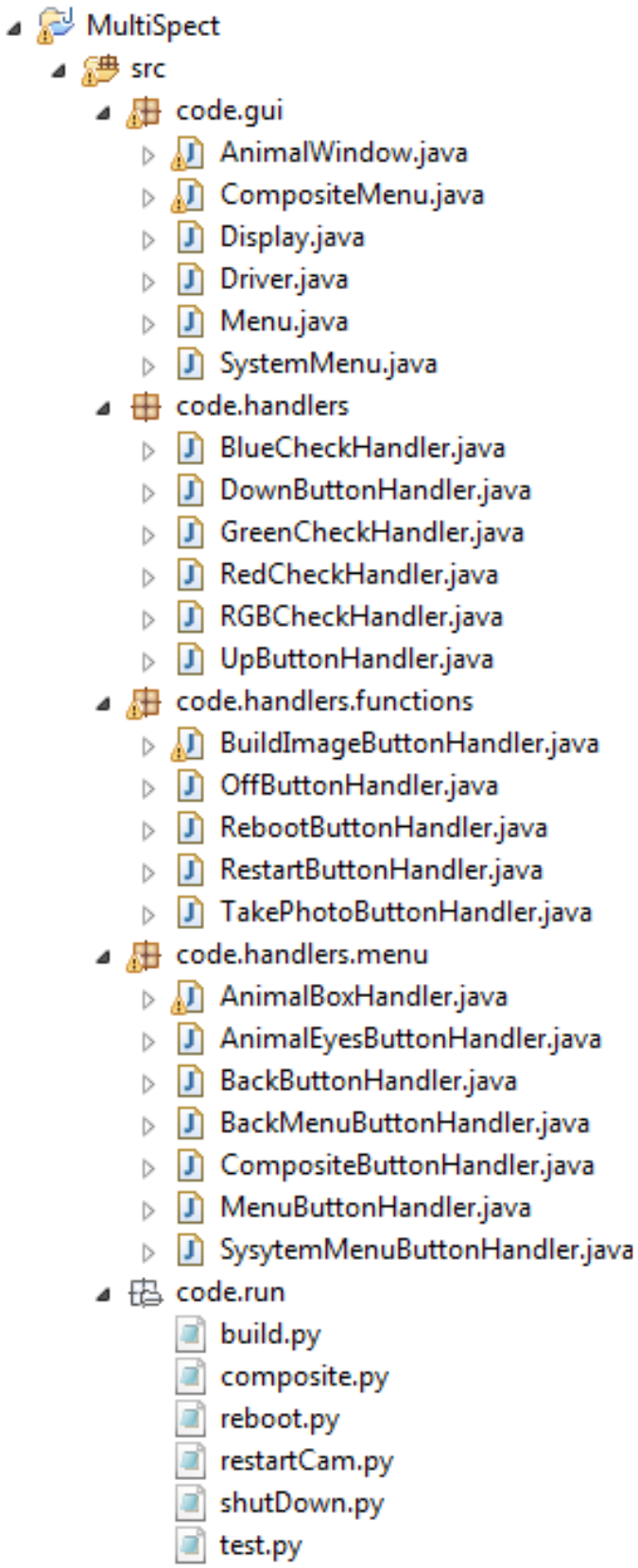

Figure 10. Image of all the classes from the Java code

(as of this writing 2e and 2f are not fully functional)

When changing lenses, the code should be edited to reflect the new names of the spectrums being analyzed to minimize confusion. The classes that hold and populate the labels are: code.handlers.functions.TakePhotoButtonHandler, code.gui.Display, code.gui.CompositeMenu, and code.gui.Menu. Be sure to also edit the labels in the resetList() function in the TakePhotoButtonHandler class. Make sure that the order for the spectra in the labels and their associated name in the names ArrayList are correlated and in the same order. After you edit this in the Java code, you have to edit the Python scripts on each Raspberry Pi to change the name that the image is saved under.

4. Acknowledgements

Thanks to Conrad Farnsworth for his part in the construction of the physical system and for the original images for those shown in Fig. 1 and Fig 2.

Thanks to Dr. Charles Tolle for his mentoring during the construction of this Multispectral Imager.

This work was made possible by the National Science Foundation award EEC-1359476 and by a grant from the Hoffert Endowment within the South Dakota School of Mines & Technology Foundation.

*These features have not been fully programmed in and require the supporting image processing script or program to be written still.