**Multiple Projects System Access Documentation**

1. **MCIS**:
   1. Physical Assembly:
      1. Ensure the unit is fully assembled with all necessary connections between the L-board, threshold, and status boards.
      2. For LED units, connect the L-board power socket to the 12V socket in the battery brick.
      3. For RFID units, connect the RFID dev board power socket to the 9V socket in the battery brick.
      4. To access system wide and MCIS code files, connect the Pi’s HDMI0 port to a monitor using an HDMI cable. Note: For Pi 4, ensure you have an HDMI connector for the new HDMI port.
   2. System-wide Code Access:
      1. After physical assembly, turn on the power button on the battery brick and wait for the login screen to appear on the monitor, displaying "pi@raspberrypi:$".
      2. Stop the MCIS software service execution by entering the command "sudo systemctl stop cascadesmobile.service" in the Pi terminal.
      3. To access the Pi system, enter the command "sudo mount -o remount -o rw /". This will prompt the login screen. Enter the following details:
         1. Username: pi
         2. Password: pi0713
      4. To access the MCIS code files and libraries, open the File Manager located in the top-left corner, go to the below locations, double-click, and select Open to open the files:
         1. Main application file: cascades.py, located in /usr/bin/cascades.py
         2. Libraries: /usr/lib/cascades/
         3. HTML file: /usr/share/cascades/player\_template.html. Right-click on this file and select Geany to open it for necessary edits. Note: If made changes, update the new revision number in this HTML file located at a line section “Firmware: x.x.x”.
      5. Make changes to the code's files and libraries and save them (Ctrl + S). Open the terminal using (Ctrl + Alt + T) or clicking at Terminal icon in upper left corner and run the command "sudo python3 /usr/bin/cascades.py" to compile, run and debug the main application file.
         1. NOTE: Above command must be entered regardless of whether you have made changes or are debugging the code.
         2. To stop the execution of the file in this case, enter Ctrl + C on the terminal.
      6. After running the following the command, if message “Error 098: Address already in use” (located in the very bottom) appears on the Terminal, this means cascades.py is still using the resources. You need to stop the cascadesmobile.service again using same command as mentioned in Step ii and run the command in Step v.
      7. If the error persists, reboot the system by entering “sudo reboot” and it gets you to the Step i. and repeat the same steps.
   3. Uploading Latest Software Changes to GitHub:
      1. Use Pi system and make sure WiFi is connected (upper right corner).
      2. Open Chrome, located in top-left corner with a blue globe and navigate to github.com/TinManSystems.
      3. Log in with the appropriate credentials.
      4. Go to the MCIS repository "MCIS\_multiCam" and create a new branch (beneath the repository name) out of the default branch with a new rev number to upload the files.
         1. For an instance, you want to upload new cascades.py file. Go to /package/usr/bin/, click on “Add file” at upper right corner and select Upload Files.
         2. Click on “choose your files”, navigate to the /usr/bin/ and select cascades.py to upload. Press green button “Commit changes” to commit the changes in the file.
         3. Follow Step 1 & 2 for files in libraries or html file under package folder.
      5. To document the changes made, you need to edit the README.md file under the new branch name. Click on README.md file and then click on “edit this file” icon at upper right corner to make edits and press on Commit changes green button to save changes.
   4. Accessing Board Files on GitHub:
      1. Recommended platform: Windows PC
      2. Open the browser, go to github.com/TinManSystems and log-in.
      3. For LED/Non-LED based MCIS unit, refer to the repository "MCIS-v3.5-board-files."
      4. For LED/Non-LED and RFID-based MCIS unit, refer to the repository "MCIS-v4.0-board-files."
      5. To download the files, get on the repository, press on “Code” button and download zip. Unzip the files to access every board files.
      6. If any changes to the board files are made, upload them to the respective repositories mentioned above. Follow upload process as explained in Step c.
   5. Auto-Software Update Process (Step-by-Step):
      1. Ensure the new revision number is updated in the following file locations on GitHub:
      2. /MCIS\_multCcam/package/usr/share/cascades/player\_template.html
      3. /MCIS\_multiCam/build-deb.sh
      4. /MCIS\_multiCam/package/DEBIAN/control
      5. Download the MCIS\_multiCam repo to the Linux laptop or directly onto the Pi.
      6. Unzip the repository to its folder name.
      7. In the same folder, open the terminal and enter the command "sudo sh build-deb.sh" to create two files named "cascades-mobile\_x.x.x.zip" and "cascades-mobile\_x.x.x.deb" in the existing repository folder.
      8. Insert the SD card used for recording sessions and create a new folder named "fw\_upd."
      9. Copy and paste the file "cascades-mobile\_x.x.x.zip" inside the "fw\_upd" folder.
      10. Turn off the power of MCIS and insert the SD card.
      11. Turn on the power and wait for the system to cycle through the LEDs on the Status/Threshold boards twice.
      12. If the second round of cycling through the LEDs is successful, the MCIS system is updated with the new revision.

1. **Ranger**:
   1. Physical assembly
      1. Ensure the unit is fully assembled with all necessary connections between the L-board, threshold, and status boards.
      2. To access system wide and MCIS code files, connect the Pi’s HDMI0 port to a monitor using an HDMI cable. Note: For Pi 4, ensure you have an HDMI connector for the new HDMI port.
   2. System-wide code access:
      1. After physical assembly, turn on the power button on the battery brick and wait for the login screen to appear on the terminal, showing "pi@raspberrypi:$".
      2. Stop the Ranger software service execution by entering the command "sudo systemctl stop amznranger.service" in the Pi terminal.
      3. To access the Pi system, enter the command "sudo mount -o remount -o rw /". This will prompt the login screen. Enter the following details:
         1. Username: pi
         2. Password: raspberry
      4. To access the Ranger code files and libraries, open the File Manager located in the top-left corner, go to the below locations, and double-click to open the files:
         1. Main application file: ind3.py, located in /usr/bin/ind3.py
         2. Libraries: /usr/lib/amznranger/
      5. Make changes to the code's files and libraries and save them (Ctrl + S). Open the terminal and run the command "sudo python3 /usr/bin/ind3.py" to compile and run the main application file.
   3. Upload latest changes to the Github:
      1. Make sure WiFi is connected.
      2. Open Chrome, located in top-left corner with a blue globe, in the Pi and navigate to github.com/TinManSystems.
      3. Log in with the appropriate credentials.
      4. Go to the Ranger repository “RV3\_singCam/package/usr” to upload the respective files.
   4. Access Board files on Github:
      1. Recommended platform: Windows PC.
      2. Refer to the repository: Ranger-v3.0-board-files.
      3. If any changes to the board files are made, upload them to this repo.
   5. Auto-software update process (step-by-step):
      1. Make sure new revision number is written to these file locations on the github:
         1. /RV3\_singleCam/package/usr/lib/amznranger/svgdisplay.py
         2. /RV3\_singleCam/build-deb.sh
         3. /RV3\_singleCam/package/DEBIAN/control
      2. Download the RV3\_singleCam repo to the Linux laptop or in the Pi itself.
      3. Unzip the repo to its folder name.
      4. In the same folder, open the terminal and enter the command “sudo sh build-deb.sh” and it will create two files with the names “amznranger\_x.x.x.zip” and “amznranger\_x.x.x.deb” in the existing repo folder.
      5. Insert the SD card used for recording sessions and create a new folder named “fw\_upd”
      6. Copy and paste the amznranger\_x.x.x.zip inside fw\_upd folder.
      7. Turn-off the power of Ranger and insert the SD card.
      8. Turn-on the power and wait for the system to cycle through the LEDs on the Status/Threshold boards twice.
      9. If 2nd round of cycling through the LEDs is successful, then it means Ranger system is updated with new rev.
2. **Gateway:**
   1. Physical assembly:
   2. System-wide code access:
   3. Upload latest changes to the Github:
   4. Access Board files on Github:
   5. Auto-software update process (step-by-step):
3. **Locate Production image files on Linux system:**
   1. In the main screen, click on “600GB Volume” icon on task bar on left hand side.
   2. Daniel-> MCIS\_multicam\_img\_files\_Pi4->Production->MCIS\_multicam\_flagship\_x.x.x.img
4. **Flash latest .img files into SD Card using Linux system:**
   1. Insert the empty SD card into the laptop.
   2. In the main screen, click on the folder Etcher and double click on the file ending with. Appimage to open it.
   3. Once the Etcher pops up on the screen, click on “flash from file” and locate the latest production file under “Other Locations” as explained above.
   4. Click on “Select Target” and select the SD card.
   5. Lastly, click on Flash! and insert the Linux password (0713)
5. **Locate latest .zip files on Linux system:**
   1. In the main screen, click on “600GB Volume” icon on task bar on left hand side.
   2. Daniel->MCIS\_multiCam-Rev-LSM9/6-LED-RFID-**x.x.x** (look for latest rev here)->cascades-mobile\_x.x.x.zip
6. **How to re-flash the SD card:**
   1. Format the SD card. Follow these steps:
      1. Use Windows PC
      2. In the search bar, type “Create and format the hard disk partition” and click on it.
      3. Look out for the SD card and delete its partitions if there are more than 1 (shows Unallocated)
      4. Allocate the memory by double-clicking on it and click New simple Volume. Click Next till the end. SD is fully formatted now to re-flash it.
7. **Calibrate units.**
   1. Assuming your Windows PC and unit are both on same Wi-Fi
   2. Make sure RTC coin cell is inserted on to its L-board slot.
   3. Power-on the unit.
   4. Open the utility tool “mobileSystemsUtility.py” in your PC.
      1. If you don’t have this file, go to Github repository “Mobile Systems Config Tool py files”, click on Code green button to download the zip file and unzip it to access the mobileSystemsUtility.py file
   5. Check the Calibration and Serial Number Assignment.
   6. click on Calibrate Sensors button to calibrate the sensors.
   7. Click on Set time button to set the time.
   8. Insert the Serial number in an input text, click on Set SN button to give the unit a Serial Number and unit will restart to make the changes permanent.
   9. Record the session to check if time is up-to-date and SN is successfully written.
   10. **Worst case scenario**: if time is still not up to date, this could mean either the RTC chip was bad, or your PC and unit were not same Wi-Fi.
       1. Power-off the unit.
       2. Connect unit’s HDMI1 port to monitor using HDMI cable.
       3. Power-on the system and wait for the login screen to appear on the monitor, displaying "pi@raspberrypi:$".
       4. To access the Pi system, enter the command "sudo mount -o remount -o rw /". This will prompt the login screen. Enter the following details:
          1. Username: pi
          2. Password: pi0713
       5. Check on the Wifi icon on upper right corner and confirm if its connected to same wifi as your Windows PC. If not, change it, wifi icon starts to toggle and wait for it to become stable.
       6. Follow step d. onwards to set the time and set the SN again to restart the unit and cycle through the LEDs to make changes permanent.
       7. Record the session to validate if the time is set accurately.