Instructions for Running the Smart Mirror Software on the Raspberry Pi

To prepare a Raspberry Pi, it is recommended that you use the basic operating system, Raspbian for best compatibility. The first order of business is to make sure the operating system is the most current version available.

Updating the Raspbian Operating System Using Terminal

1. Open up terminal (Insert Home screen with arrow pointing to terminal location Image HERE)
2. Make sure you are in the root directory by typing: cd ~
3. After you have confirmed you are working in the root directory, input this command: sudo apt-get update
    a. The first part of this command is sudo. Sudo is a way of telling the machine to run the command as an administrative command.
    b. The next command is apt-get. Apt-get tells the computer we are fixing to, find whatever comes next, and then install whatever we have just gotten.
    c. Update is the newest/current version of the Raspbian operating system
4. The next command is: sudo apt-get upgrade
    a. The upgrade part of this command simply updates what packages are current and their versions for the operating system.

Installing SDL2 Using Terminal

1. The first command to run here is like updating the operating system. Here we once again confirm we are working in the root directory by running: cd ~
2. Once we have confirmed we are in the root directory, we then execute the command: wget https://www.libsdl.org/release/SDL2-2.0.3.tar.gz
    a. Wget is a command that follows a link and downloads whatever the link points to, if it is a file or package.
3. We then unzip the package with the following command: tar zxvf SDL2-2.0.3.tar.gz
    a. Tar is a command that unpacks files ending with .tar.gz
4. After we have unpacked the file, we must then create a directory for the contents: cd SDL2-2.0.3 && mkdir build && cd build
    a. Cd moves to the directory with the name given after cd and updates the current directory to that directory
    b. Mkdir creates a directory with the name given after the command
    c. Cd build moves us to the directory build. To recap, we are now working in ~/SDL2-2.0.3/build
5. After we have unpacked all of the files we must then configure the build: ../configure --disable-pulseaudio --disable-esd --disable-video-mir --disable-video-wayland --disable-video-x11 --disable-video-opengl
    a. ../configure configures each of the features of the library being installed with their build flags for the Pi.
6. Now we must now configure a make file and build it: make -j 4
7. We must install the make file we just created: sudo make install
8. SDL2.0.3 has now been installed on the machine

Installing Additional SDL2 libraries (SDL_image & SDL_ttf) using Terminal

To configure additional libraries, we must follow similar steps to installing SDL2.

1. To install SDL_image we must first move the current working directory to the root directory: cd ~
2. We must then download the SDL_image files: wget http://www.libsdl.org/projects/SDL_image/release/SDL2_image-2.0.0.tar.gz
3. We need to unzip the files next: tar zxvf SDL2_image-2.0.0.tar.gz
4. We now need to move the current working directory to the file we just unzipped and create build directory within the SD2_image directory: cd SDL2_image-2.0.0 && mkdir build && cd build
5. The files must be configured so we can install them: ../configure
6. Now we need to make the make files to be installed: make -j 4
7. Now we need to install the make files to the pi: sudo make install

Installing SDL_ttf Library using Terminal

1. To install SDL_ttf we must first move to the current working directory again: cd ~
2. Now we must download the SDL_ttf files: wget http://www.libsdl.org/projects/SDL_ttf/release/SDL2_ttf-2.0.14.tar.gz
3. We need to once again unzip the files: tar zxvf SDL2_ttf-2.0.14.tar.gz
4. Now we need to move the current working directory to the file we just unzipped and create a build directory within the SLD2_ttf directory: cd SDL2_ttf-2.0.14 && mkdir build && cd build
5. The files must be configured so we can install them: ../configure
6. To make the files ready to be installed, we must make the files to be installed: make -j 4
7. Now we just need to install the make files to the pi: sudo make install

Installing SQLite3 using Terminal

To install SQLite3, we need to follow similar steps as we did with SDL2. SQLite3 allows us to have basic functionality with the database onboard of the Pi.

1. First we must move the current working directory to the root directory: cd ~
2. Now we must download the SQLite3 c source code as an amalgamation: wget http://www.sqlite.org/2017/sqlite-autoconf-3180000.tar.gz
3. We must now unzip this file: tar zxvf sqlite-autoconf-3180000.tar.gz
4. Now we must move the current working directory to the file: cd sqlite-autoconf-3180000
5. We need to let the built in configure configure sqlite3 for the system: ./configure
6. To install the package we have to create a make file for it: make -j 4
7. Now we must install that make file: sudo make install

Installing Weather Utility on Raspberry Pi using Terminal

In order for the software to obtain the weather information for your location, we must first do a couple of steps to enable this feature.

1. Firstly we must install the weather utility: sudo apt-get install weather-util
2. Now we must move the current working directory to: sudo cd /usr/share/weather-util
3. We must then remove the contents that are currently there: sudo rm –f stations.gz zones.gz airports.gz
4. Now we must get the weather info needed for our location: sudo wget fungi.yuggoth.org/weather/src/weather/stations \fungi.yuggoth.org/weather/src/weather/zones \fungi.yuggoth.org/weather/src/weather/airports
5. Now weather utility should be functioning on the machine

Compiling The Software

To compile the software, after you've downloaded it from the GitHub folder named Project, you must navigate your current working directory to where you placed the software folded on your pi.

1. There are a couple of steps that need to be taken before the software is compiled.
    a. First we must make sure that the c++ compiler is on the pi by doing: g++ -v
    b. If g++ is installed and shows a version then there is no further action needed
    c. If g++ is not installed on the system, you must do: sudo apt-get install gcc
2. For me, the software's folder was on the Desktop: cd Desktop/Project
3. Now we must compile the program before we run it: $g++ -std=c++0x Source.cpp Basic_Image.cpp Clock.cpp Text.cpp Widget.cpp TextDatabase.cpp Weather.cpp –o Source $(pkg-config --libs –-cflags sdl2 SDL2_image SDL2_ttf sqlite3)
4. If no warnings or errors were given: run the software by doing: ./Source
    a. If there is no keyboard or mouse input try installing: sudo apt-get install libudev-dev
    b. After the dev library has been installed, (You may need to restart the Pi before following these steps) check to make sure it is enabled:
        i. cd ~
        ii. cd SDL2_image-2.0.0
        iii. ./configure
        iv. Search for the line saying: Using libudev : YES
        v. If it says that libudev is enabled, restart the Pi and then compile the software and run it.
        vi. If not, restart the Pi and follow the steps i-iii above.