Building the EmulationCore on a Raspberry Pi 3

EmulationCore is the "prototype" of the Voice Control System for IVI.

The EmulationCore was developed on a Windows PC, so the initial configuration was done to fit the Windows Environment. In order to setup the project inside Raspbian Environment, you may follow the instructions below.

Repository Folder: https://github.com/GENIVI/GENIVI-GSoC-18/tree/master/EmulationCore

**Prerequisites**

- Debian based Linux distro (raspbian) installed on your RPi.
  - These instructions have been tested on:
    - RPi Desktop
    - Linux raspberrypi3 4.9.35-v7+ #1014 SMP armv7l GNU/Linux
  - Find out more here: https://www.raspberrypi.org/downloads/

**Installation of Python 3.6.5**

- This is the point at which the installation becomes different from the installation on a Debian system like Ubuntu.
- Python 3.6.5 isn't available via the APT(Advanced Package Tool) as a package, but can be installed by directly downloading the sources and building it on our RPi.
- You may follow the following set of instructions to get Python 3.6.5 installed.

**Steps**

1. Pull up a terminal on your RPi,
2. Update your packages with the repositories followed by an upgrade.

```bash
sudo apt-get update
sudo apt-get upgrade
```

3. You need C/C++ build tools for linux. Use the following command to grab them.

```bash
sudo apt-get install build-essential checkinstall
```

4. You also need some dependencies to build Python 3.6.5 in your RPi, just run the following command and grab them. This might take a while to install.

```bash
sudo apt-get install libreadline-gplv2-dev libncursesw5-dev libssl-dev libsqlite3-dev tk-dev libgdbm-dev libc6-dev libbz2-dev
```

5. Navigate to src directory of usr folder and download the Python 3.6.5 source into that folder. Extract the tarball. Navigate into the extracted folder.

```bash
cd /usr/src
sudo wget https://www.python.org/ftp/python/3.6.5/Python-3.6.5.tgz
sudo tar xzf Python-3.6.5.tgz
cd Python-3.6.5
```

6. Compile the code with the aid of build-tools.

```bash
sudo -s
bash configure
./configure --enable-optimizations (optional)
makedist
make install
exit
```

7. Now you're good to go! Run the following command on your terminal to confirm your installation. This should just print back the Python version as 3.6.5.

```bash
cd /usr/src
```
Note #1
If you build Python 3.6.5 from source, you should have Python-Dev Tools. Therefore you do not need to install them separately.

Note #2
Usually Raspbian comes with Python Virtual Environment Tools and Pip3 Python Package Management Tools pre-installed. Therefore you would not need to install them manually. In case they don’t exist, run the following commands to get them installed.

```bash
sudo apt install python3-venv
sudo apt-get -y install python3-pip
```

Other Driver Requirements

1. **Speech Engine: espeak for Linux**
   a. Raspbian does not seem to have a default speech engine installed.
   b. We would certainly need a Speech Engine to output data to the user as mentioned in the library we've used: pyttsx3
   c. Since RPi's Raspbian is a debian linux distro, we would need espeak.
   d. Run the following command to get espeak speech engine installed.

   ```bash
   sudo apt install espeak
   ```

2. **FLAC Encoder**
   a. In order for speech recognition to work, we would need an encoder. This is purely a library dependency.
   b. According to the documentation of the library we've used, we would need FLAC encoder.
   c. Since this is required only if the system is not x86-based Windows/Linux/OS X (as instructed in documentation of the library), we should go ahead and install it.
   d. Run the following command to get FLAC installed.

   ```bash
   sudo apt install flac
   ```

3. **Driver Extensions, Audio Stream Handling libraries and other Developer Extensions**
   a. We would need certain libraries and extensions on the system side for TTS and STT to perform their job.
   b. Run the following command to get them installed.

   ```bash
   sudo apt install libasound-dev portaudio19-dev libportaudio2 libportaudiocpp0 ffmpeg libav-tools
   ```
   
   (NB: this may be libasound2-dev on your machine. Also, ffmpeg may be called libav in Raspbian.)

4. **Dependencies of Processes**
   a. Similar to the above dependencies being related to EmulationCore, there may be some dependencies required to be installed by the Processes.
   b. Please follow the guide here: Prerequisites for Processes to get them installed.
   c. Return back here once you've completed installing them.

5. **Dependencies of Task Executors**
   a. Similar to the above dependencies being related to EmulationCore, there may be some dependencies required to be installed by the Task Executors.
   b. Please follow the guide here: Prerequisites for Task Executors to get them installed.
   c. Return back here once you've completed installing them.

Cloning the Project

- Navigate into the folder where you need to clone the project in a terminal.

  ```bash
  cd "/path/to/folder/to/clone"
  ```

- To clone the repository, just run the following command.
git clone https://github.com/GENIVI/GENIVI-GSoC-18.git

- **Migration/CMUSphinx:** If you need to continue with CMUSphinx for Speech Recognition, follow the instructions in this page.
- Wait until everything's completed.

**Creating the Virtual Environment**

- We must create our virtual environment inside the EmulationCore folder of the folder you've cloned the project.
- Navigate into the EmulationCore directory in the directory you've cloned the entire repository.

  ```
cd */path/to/folder/to/clone/EmulationCore*
  ```

  or if you're currently inside the cloned directory,

  ```
cd EmulationCore
  ```

- Initialize a virtual environment as follows.

  ```
  python3.6 -m venv python_venv
  ```

- Activate the virtual environment

  ```
  source python_venv/bin/activate
  ```

**Installing the Dependencies**

**Packages from PyPI**

- Some of the requirements are specified inside the requirements.txt in EmulationCore folder.
- Just run the following command in the activated virtual environment (in terminal) to get them installed.

  ```
  pip3 install -r requirements.txt
  ```

- Note that every dependency will be installed only for the virtual environment, so they won't be globally available.

**Packages need to be Compiled**

- In order to install other requirements which might be need to be compiled under own environments, please follow the guides:
  1. Dependencies for Processes.
  2. Dependencies for Task Executors.
- Please note that this step is also vital for the EmulationCore to work as it would contain must-have set of dependencies for it work.

**Installing NLTK Tokenizer**

- We need to install a word tokenizer for NLTK library.
- NLTK has a set of tokenizers, we're supposed to use "punkt" as our tokenizer.
- Run the following commands to get it installed.
- Note: This installation may take a while.
- First get a python console in your terminal (inside activated virtual environment).

  ```
  python3.6
  ```

- Import nltk and download the punkt tokenizer

  ```
  import nltk
dlkt.download("punkt")
  ```

- Get rid of the python console by typing the following command, which returns you back to the bash.
Changing the Configuration

Please follow the instructions given here to change the configuration of EmulationCore.

Note: This must be done, otherwise EmulationCore would cease to function properly.

Testing the Program

- Just run the following command from the EmulationCore directory, in activated virtual environment to get the system running.

  python3.6 base/run.py

Note #3

If you are experiencing issues with sound output from the RPi, please consult the forums, specially threads like this.

Having no sound output is a system-wide issue. This is due to some HDMI devices connected to the RPi, not only just by the program itself.

There will be no instances where system sounds of RPi remains working and EmulationCore sounds failing to work.

If you've completed every step correctly, program will run fine.

Please note that these instructions are only for Raspbian on RPi system, and NOT for all linux systems. The functionality would not be guaranteed on other systems currently.

Good Luck!😊