FRANKFURT UNIVERSITY OF APPLIED SCIENCES

PROJECT MANUAL

Automotive Sensor for Object Recognition using Red Pitaya and Raspberry Pi (SL-2)

Autonomous Intelligent Systems

M.E. in Information Technology

Winter Semester 2019/20

Group ID : SL2-10

Devan Nair
1274369

Prashant Pratik
1271359

Under the Guidance of :

Prof. Dr. Peter Nauth
Mr. Julian Umansky
Mr. Michalik

1. To establish ssh connection between Red Pitaya and Raspberry Pi :

1.1. Create a shell script “{proj-name-ssh-script}.sh” on Raspberry Pi using “sshpass” command.
1.2. Use username, password and IP address of Red Pitaya in “sshpass” command.
1.3. Enable executable rights to the user for this script file.
1.4. Execute this shell script. This will establish the required ssh connection.

2. To start Ultrasonic Sensor on Red Pitaya :

2.1. Establish ssh connection to Red Pitaya. This will execute “iic.c” code on Red Pitaya.
2.2. On execution of “iic.c” code, Ultrasonic Sensor on Red Pitaya will start sending data.

3. To send SCPI commands from Raspberry Pi for data acquisition, feature extraction and classification :

3.1. Establish ssh connection to Red Pitaya.
3.2. Execute “daq.py” python code from Raspberry Pi.
3.3. This “daq.py” is using “redpitaya_scpi.py” to access Red Pitaya over the IP network using SCPI commands.
3.4. Further SCPI commands written in “daq.py” is used to plot the real time signal data coming from Red Pitaya.
3.5. For feature extraction, execute “signalFeaturePersistence.py” code to save and plot both time and feature representation of the real time signal data coming from Red Pitaya.
3.6. Execute “liveEstimate.py” code for real time classification of 3 objects namely wall, car and human. This classification is displayed by 3 different LEDs on Raspberry Pi.

4. To enable plug-n-play :

4.1. Create a shell script “{proj-name-script}.sh” on Raspberry Pi that will execute shell script “{proj-name-ssh-script}.sh” for ssh connection and then, will execute python code for real time classification.
4.2. Create a service daemon description describing this shell script to be called after boot-up network initialization.
4.3. Move the service daemon file to “/etc/systemd/system”. Save it as “{proj-name}.service”.
4.4. Enable executable rights to the user for the service file.
4.5. Execute “systemctl start {proj-name}.service”.
4.6. Make sure there are no errors in the resulting log file. If there are any fix them.
4.7. Execute “systemctl enable {proj-name}.service” to enable daemon startup at boot-up.

5. Packages to be installed on Raspberry Pi :

xrdp, sshpass, tensorflow, tsfresh

6. Scripts and Python codes on Raspberry Pi :

sshscript.sh, projectscript.sh, classify.service, daq.py, redpitaya_scpi.py, signalFeaturePersistence.py, liveEstimate.py, lstm_model.h5