Qualcomm Developer Project smarttraffic-demo2.0

Project Submission

|  |  |  |
| --- | --- | --- |
| **Attribution:** |  | |
| **Email address** | [zhangzz6687@thundersoft.com](mailto:zhangzz6687@thundersoft.com)  [zhanglei0706@thundersoft.com](mailto:sunzhen@thundersoft.com)  Yuandk0305@thundersoft.com | |
| **Project Title**\* | smarttraffic-demo2.0 | |
| **Images**  *Upload up to 5 images of your project*  *Please submit/send the original JPEG/PNG files for all images included in the document* | **Qualcomm® QCS610 SoC.png**    [Alt tag: “smarttraffic-demo2.0 using The Qualcomm® QCS610 SOC Open Kit ”]  **usb.png**   |  | | --- | | **typc** |   [Alt tag: “using the USB line to develop on Qualcomm® QCS610 SOC development board” ]  **dpline.jpg**  dpline  [Alt tag: “using the universal DP line to connect LED displayer to Qualcomm® QCS610 SOC development board”]  **LED-Displayer.png**  LED-Displayer  [Alt tag: “using a LED Displayer to display the smarttraffic-demo2.0 interface from Qualcomm® QCS610 SOC development board”]  **charger.jpg**  charger  [Alt tag: “using round-hole charger to power Qualcomm® QCS610 SOC development board”] | |
| **Description**\*  *High level description of the project* ***(75 words or less)*** | Using the Qualcomm® QCS610 SOC Open Kit and IMX415 camera module. Then, its powerful computing ability will been show by the license plate recognition process is completed that combine with Qualcomm Neural processing SDK for AI, OpenCV, SVM and Gstreamer.The EasyPR license plate recognition engine is transplanted to the Qualcomm® QCS610 SOC development board to display the rich application scenes of Qualcomm® QCS610 SOC.At the same time, UI is developed UI by GTK to display the progress of license plate recognition. | |
| **Objective**   * *What inspired you to create this project?* * *What is your desired outcome?* | Show the Qualcomm® QCS610 SOC practical application ability and usage scenarios. | |
| **Materials Required / Parts List / Tools** | Part Name | Link to purchase |
| Qualcomm® QCS610 SOC Open Kit | https://www.thundercomm.com/app\_zh/product/1593776185472315 |
| USB line | https://item.jd.com/40759941966.html |
| DP line |  |
| LED Displayer |  |
| Charger |  |
|  |  |
|  |  |
| **Source Code / Source Examples / Application Executable**  *Link to open source / shareable code repository* | Description | Link |
| [Source Code](https://github.com/canyudeguang/Home_Automation) | [https://github.com/ThunderSoft-XA/C610-smar](https://github.com/ThunderSoft-XA/demo-Smart-Motion-detector)ttraffic-demo2.0 |
|  |  |
|  |  |
|  |  |
| **Additional Resources**  *List related links or resources such as websites, videos, presentations, or other materials* | Resource Title | Link or File Name (and provide file) |
|  |  |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Build / Assembly Instructions** | Sample outline:   1. Overall design framework and Test environment construction method.      1. Software Build Instructions    1. Prepare a PC (Ubuntu 16.04) with Yocto Project;    2. Install adb ;    3. Configure the compilation environment according to the release note document    4. Configure the DP display environment    5. Write BB file(<traffic\_0.1.bb>),Executing the “bitbake smarttraffic-demo2.0” command generates an “smarttraffic-demo2.0” executable file 2. Start smarttraffic-demo2.0    1. Copy “smarttraffic-demo2.0” executable file to project root directory, then “adb push ” whole project file to Qualcomm® QCS610 SOC,ex./data/ dir.    2. Execute adb root && adb shell. Next, enter the smarttraffic-demo2.0 project root directory, execute ./weston\_dp\_client smarttraffic-demo2.0. | |
|  | Sample outline:   1. How does it work?   Below are some usage instructions to test the project. Now let's introduce the smarttraffic-demo2.0’s workflow. Initialize the program running environment, and then initialize the camera, Qualcomm Neural processing SDK for AI environment, and GTK UI. After the camera gets the video stream, it will be displayed by the UI, and license plate recognition will be carried out.    int main(int argc,char \*\*argv)  {  /\* start up malloc trace record log \*/  setenv("MALLOC\_TRACE", "mtrace.log", 1);  mtrace();  GMainLoop \*loop;  loop = g\_main\_loop\_new(NULL, FALSE);  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  /\* start up g\_object thread for refresh thread across thread\*/  if ( !g\_thread\_supported() ) {  #ifdef DEBUG  printf("[%s]--Init g\_thread./r/n", \_\_FUNCTION\_\_);  #endif // DEBUG  g\_thread\_init(NULL);  gdk\_threads\_init();  }  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  /\* gtk gdk gst init \*/  gtk\_init(&argc,&argv);  gdk\_init(&argc,&argv);  gst\_init(&argc,&argv);  RefreshObject \*myObject;  RefreshData \*myRefreshData;  again :  #pragma omp parallel  #pragma omp section  {  #pragma omp section  myObject->dstUI = new EasyUI();  myObject->dstUI->createUI();  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  myObject->dstUI->buildSignal("recognition-signal");  myObject->dstUI->buildSignal("refresh-signal");  myObject->dstUI->showUI();  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  #pragma omp section  snpecheck = new SNPEPlate();  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  snpecheck->init();  #pragma omp section  myObject->srcCamera = new Camera();  do {  if(myObject->srcCamera->checkElements()) {  myObject->srcCamera->setElementsPro();  if(myObject->srcCamera->linkPipeline()) {  assert( myObject->srcCamera->setPipelineStatus(GST\_STATE\_NULL) != false);  }  }  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  } while ( !myObject->srcCamera->flags);  }  if(myObject->srcCamera->flags == false || myObject->dstUI->flags == false || snpecheck->flags == false) {  goto again;  }  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  #pragma omp parallel for ordered  {  g\_signal\_connect(myObject->srcCamera->sink,"new-sample",  G\_CALLBACK(on\_new\_sample),myObject);  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  #pragma omp ordered  g\_signal\_connect(myRefreshData,"recognition-signal",  G\_CALLBACK(plate\_recognize),&myObject->imgMat);  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  g\_signal\_connect(myObject->dstUI->window,"refresh-signal",  G\_CALLBACK(refreshUI),myRefreshData);  }  myObject->srcCamera->signalHandle();  assert (myObject->srcCamera->setPipelineStatus(GST\_STATE\_PLAYING) != false);  cout << \_\_FILE\_\_ << "====" << \_\_LINE\_\_ << endl;  g\_main\_loop\_run(loop);  gdk\_threads\_enter();  gtk\_main ();  gdk\_threads\_leave;  myObject->srcCamera->~Camera();  return 0;  } | |
| **Usage Instructions** | The identification results are as follows： | |
| **Contributor(s) Info**  *Feel free to include headshots!* | Name | Title  Company |
|  |  |
|  |  |
|  |  |

––– Continued on next page –––

Filters and Tags for QDN projects page

|  |  |  |
| --- | --- | --- |
| **Platform/Hardware** | CSR 101x/102x Bluetooth  DragonBoard 410c  mangOH Red/Yellow  √ Qualcomm QCS610 SoC | MDM920x LTE for IoT  QCA-402x WiFi/BLE/Zigbee  Qualcomm Robotics RBx Dev Kit |
| **Software Tools** | 3D Audio Plugin for Unity  Adreno GPU SDK  Hexagon DSP SDK | √ Neural Processing SDK for AI  　Snapdragon Profiler |
| **Operating System** | Android  √ Linux  ThreadX RTOS | Ubuntu Core  Windows 10 IoT Core |
| **Cloud Services/Platform** | Sierra Wireless AirVantage  Gizwits Cloud Platform  AT&T M2X  IBM Bluemix | IBM Watson IoT  Microsoft Azure IoT  Amazon AWS IoT |
| **Skill Level Required** | Advanced  Beginner  √ Intermediate |  |
| **Areas of Focus** | 3D Printing & Modeling  Alexa Voice Service  √ Artificial Intelligence  Bluetooth  √ Computer Vision  Digital Signage  Education  √ Embedded  Gaming | Healthcare  √ IoT  √ Robotics  √ Security  Sensors  √ Smart Cities  Smart Home  Toys |

*By submitting your content (“Submission”), you are granting Qualcomm a royalty-free, perpetual, non-exclusive, unrestricted, worldwide license to: (a) post, use, copy, sublicense, adapt, transmit, publicly perform or display any such Submission, (b) use, reproduce, modify, adapt, publish, translate, create derivative works from, distribute, perform, play, host, communicate, make available and publish your Submission without restriction and (c) sublicense to third parties the unrestricted right to exercise any of the foregoing rights granted with respect to the Submission. The foregoing grants shall include the right to exploit any ideas, concepts, intellectual property, or proprietary rights in such Submission, including but not limited to rights under copyright, trademark, servicemark or patent laws under any relevant jurisdiction without Qualcomm owing any monies to you whatsoever. You represent and warrant that you own all right, title and interest in and to the Submission, or you have been granted sufficient rights in and to the Submission allowing the foregoing use of such Submission.*