Qualcomm Developer Project SmartTraffic

Project Submission

|  |  |  |
| --- | --- | --- |
| **Attribution:** |  | |
| **Email address** | [zhangzz6687@thundersoft.com](mailto:zhangzz6687@thundersoft.com)  [zhanglei0706@thundersoft.com](mailto:sunzhen@thundersoft.com) | |
| **Project Title**\* | **SmartTrffic** | |
| **Images**  *Upload up to 5 images of your project*  *Please submit/send the original JPEG/PNG files for all images included in the document* | **C610.png**   **[alt tag: “SmartTraffic using The TurboX C610 Open Kit** ”] **usb.png**   |  | | --- | | **typc** |   [alt tag: “**using the usb line to develop on turbox C610 development board.** ”] | |
| **Description**\*  *High level description of the project* ***(75 words or less)*** | Using The TurboX C610 Open Kit, combined with snpe, opencv and SVM, the localized license plate recognition process is completed to show its powerful computing power.The easypr license plate recognition engine is transplanted to the c610 development board to display the rich application scenes of c610. | |
| **Objective**   * *What inspired you to create this project?* * *What is your desired outcome?* | Show TurboX C610 practical application ability and usage scenarios. | |
| **Materials Required / Parts List / Tools** | Part Name | Link to purchase |
| TurboX C610 Open Kit, | https://www.thundercomm.com/app\_zh/product/1593776185472315 |
| USB line | https://item.jd.com/40759941966.html |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **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-demo |
|  |  |
|  |  |
|  |  |
| **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.   c610   1. Software Build Instructions    1. Prepare a PC (Ubuntu 16.04);    2. Install adb ;    3. Configure the compilation environment according to the release note document    4. Write BB file(<smarttraffic\_0.1.bb>),Executing the “bitbake smarttraffic” command generates an “smarttraffic” executable file 2. Start SmartTraffic    1. Copy “smarttraffic” executable file to smarttraffic project root directory,then “adb push ” whole project file to TurboX C610,ex./data/ dir.    2. Adb root && adb shell.Next,enter the smarttraffic project root directory,execute ./smarttraffic. | |
|  | Sample outline:   1. How does it work?   Below are some usage instructions to test the project.Now let's introduce the SmartTraffic ’s workflow.It uses the typical process of license plate recognition, the general process is license plate location, license plate correction, character extraction, character recognition.  smarttraffic/src/core/plate\_recognize.cpp  // main method, plate recognize, contain two parts  // 1. plate detect  // 2. chars recognize  int CPlateRecognize::plateRecognize(const Mat& src, std::vector<CPlate> &plateVecOut, int img\_index) {  // resize to uniform sizes  float scale = 1.f;  Mat img = uniformResize(src, scale);  // 1. plate detect  std::vector<CPlate> plateVec;  int resultPD = plateDetect(img, plateVec, img\_index);  if (resultPD == 0) {  size\_t num = plateVec.size();  for (size\_t j = 0; j < num; j++) {  CPlate& item = plateVec.at(j);  Mat plateMat = item.getPlateMat();  SHOW\_IMAGE(plateMat, 0);  // scale the rect to src;  item.setPlateScale(scale);  RotatedRect rect = item.getPlatePos();  item.setPlatePos(scaleBackRRect(rect, 1.f / scale));  // get plate color  Color color = item.getPlateColor();  if (color == UNKNOWN) {  color = getPlateType(plateMat, true);  item.setPlateColor(color);  }  std::string plateColor = getPlateColor(color);  if (0) {  std::cout << "plateColor:" << plateColor << std::endl;  }  // 2. chars recognize  std::string plateIdentify = "";  int resultCR = charsRecognise(item, plateIdentify);  if (resultCR == 0) {  std::string license = plateColor + ":" + plateIdentify;  item.setPlateStr(license);  plateVecOut.push\_back(item);  if (0) std::cout << "resultCR:" << resultCR << std::endl;  }  else {  std::string license = plateColor;  item.setPlateStr(license);  plateVecOut.push\_back(item);  if (0) std::cout << "resultCR:" << resultCR << std::endl;  }  }  if (getResultShow()) {  // param type: 0 detect, 1 recognize;  int showType = 1;  if (0 == showType)  showDectectResults(img, plateVec, num);  else  showDectectResults(img, plateVecOut, num);  }  }  return resultPD;  } | |
| **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  √ Turbox C610 | 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.*