Qualcomm Developer Project SmartCashierApp

Project Submission

|  |  |  |
| --- | --- | --- |
| **Attribution:** |  | |
| **Email address** | [zhangzz6687@thundersoft.com](mailto:zhangzz6687@thundersoft.com)  [zhanglei0706@thundersoft.com](mailto:sunzhen@thundersoft.com) | |
| **Project Title**\* | **SmartCashierApp** | |
| **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: “SmartCashierApp using The TurboX C610 Open Kit** ”] **usb.png**   |  | | --- | | **typc** |   [alt tag: “**using the usb line to develop on turbox C610 development board.** ”]  **451module.jpeg**  **415module** | |
| **Description**\*  *High level description of the project* ***(75 words or less)*** | UsingThe TurboX C610 Open Kit, the data of 1080 cameras are collected by local files , decoded and save to xxx.mp4 file.Then,using gstreamer capture picture from previous video file.These pictures will be identified to get goods information. | |
| **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 |
| 415module |  |
|  |  |
|  |  |
|  |  |
|  |  |
| **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)tcashier-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.      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(<SmartCashierApp.bb>), compile the executable file of SmartCashierApp into the system image, and burn the system according to the realease note document. 2. Start SmartCashierApp   Start the SmartCashierApp according to the 《Turbox-C610-SmartCashierApp\_User Guide》document | |
|  | Sample outline:   1. How does it work?   Below are some usage instructions to test the project.Now let's introduce the SmartCashierApp ’s workflow. It can capture images of objects and identify them with a camera.  setCameraEnv---->CreateCameraPipeline---->CreateImgPipeline---->getImg---->getResult  SmartCashier/test/SmartCashier.cpp  int main(int argc, char \*argv[])  {  string inputInfo;  if(setCameraEnv()) {  printf("camera env init failed\n");  return -1;  }  system("source /etc/gstreamer1.0/set\_gst\_env.sh");  GstCamera \*myGstCamera = new GstCamera();  DetProcess \*myDetPro = new DetProcess();  GstImg \*myGstImg = new GstImg();  myGstCamera->CreateCameraPipeline();  myGstCamera->setPipelineState(GST\_STATE\_PLAYING);  cout << "Please put the goods...." << endl;  cout << "please confirm that all goods been scanned or not??(Y or N):" << endl;  cin >> inputInfo;  myGstCamera->sendEventAndGetInfo(gst\_event\_new\_eos());  cout << "enter img pipeline" << endl;  myGstImg->CreateImgPipeline();  //myGstCamera->VideoToImg(imgpath);  cout << "start detection goods...." << endl;  int i = 0;  long start\_pre = getCurrentTime\_ms();  while(myDetPro->getImg(i) == 0) {  myDetPro->getResult();  i++;  }  long diff\_sum = getCurrentTime\_ms() - start\_pre;  std::cout << "sum cost time=" << diff\_sum << std::endl;  /\*test static orange.jpeg\*/  //myDetPro->getStaticImg();  //myDetPro->getResult();  i = 0;  cout << "goods info as follow:" << endl;  int price = 0;  for(i; i < myDetPro->goodsTotal; i++) {  cout << myDetPro->goodsNameSet[i] << endl;  int nPos = myDetPro->goodsNameSet[i].find("$");  if(nPos != -1)  {  string str = myDetPro->goodsNameSet[i].substr(nPos+1, myDetPro->goodsNameSet[i].size());  price += atoi(str.c\_str());  }  }  cout << "total price: $" << price << endl;  myDetPro->~DetProcess();  system("rm /data/detection/img/image\*");  return 0;  } | |
| **Usage Instructions** | The identification results are as follows：  download | |
| **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.*