



社区

社区 无线安全 软件无线电 [LimeSDR] Made Simple 8 C语言范例和SoapySDR API

发新帖

返回列表

查看: 678 | 回复: 2

jamesshao8



10	23	80
主题	帖子	积分

注册会员



积分 80

发消息

[LimeSDR] Made Simple 8 C语言范例和SoapySDR API [复制链接]



发表于 2018-12-22 11:45:41 | 只看该作者

楼主 电梯直达



这是第八篇教程。我们在第一篇就说过，我们会从SDR新手开始一步步教你，直到你学会调用API编程。前几篇说完后，现在是时候开始写程序了。如果你已经读完了前面的文章，会用GNU Octave，也熟悉Pothos和GNU Radio了，那么直接调用API只需要再前进一小步。

在Linux下开发(Ubuntu系统)

首先我们需要一个编译器，我们使用gcc，它是Linux的主要部件，所以你很可能已经有了。如果你没有，可以运行sudo apt-get install gcc来安装最新版本。

搞定后，我们来试试第一个Hello World程序。

创建一个hello_world.c，用你最喜欢的编辑器打开文件，加入下面的代码：

```
#include // so we can use printf
int main()
{
    printf("\nhello world\n"); // "\n" means new line
}
```

保存并关闭。接下来我们需要使用gcc编译它，我们运行这个命令： gcc -std=c99 hello_world.c -o helloworld.bin
这个命令会生成helloworld.bin，它是一个可执行程序，如果你没有写-o，那么会自动生成a.out。-std=c99表示使用的c语言的标准。我们接下去使用的soapy例子就是c99标准。

运行helloworld.bin会得到如下输出：

```
karl@bronte: ~/Documents/LimeSDR/Soapy_API_Work
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$ gcc -std=c99 helloworld.c -o helloworld.bin
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$ ./helloworld.bin
hello world
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$
```

SoapySDR接口(API)
方便起见, 我们可以把SoapySDR API看作一个黑盒。就好像printf一样, 我们只需会用, 当我们填入正确的数据后, 它会按照我们的要求来工作。这个API有c++和python版本, 你如果不熟悉c语言也可以使用另两个版本。

API文档在Device.h这个头文件里。
SoapySDR的网站上有例子, C语言的例子在这里。
要使用这个API, 必须安装SoapySDR。推荐你用Linux, 但是Windows也是支持的, 只不过设置编译器会更复杂。如果你之前也看了我的教程, 那么你已经装过了, 因为我们之前已经用了很多次SoapySDR了。如果没有安装, 可以点这里。

你可以下载C语言的例子, 并且保存为example.c文件。
然后就想编译helloworld.c一样, 只不过这次gcc命令要这样写:
\$ gcc -std=c99 example.c -lsoapySDR -lm -o example.bin
这次我们需要加入链接命令 -lm, 这是因为要用到一些数学上的函数, 我们最好每次都加上这个, 这样不容易出错。

然后运行一下, 会报错: SoapySDRDevice_make fail: RTL-SDR device not found.

但是这个例子并不要 RTL-SDR。
不要害怕, 只需要做些修改, 这个代码就适合 LimeSDR 了, 把如下文字改一下:
"SoapySDRKwargs_set(&args,"driver", "rtlsdr");"
改为:
SoapySDRKwargs_set(&args,"driver", "lime");
编译, 再次运行, 你就会看到如下显示:

```
karl@bronte: ~/Documents/LimeSDR/Soapy_API_Work
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$ gcc -std=c99 example.c -lsoapySDR -lm -oexample.bin
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$ ./example.bin
Found device #0: Addr=1d503c108 driver=Lime label=LimeSDR-USB [USB 2.0] 9660804/2118, meta=USB 2.0, module=5TREAN, name=LimeSDR-USB, serial=966080472118
[INFO] Make connection: 'LimeSDR-USB [USB 2.0] 966080472118'
[INFO] Device name: LimeSDR-USB
[INFO] Reference clock: 30.720 MHz
[INFO] Init: LMS7002M(0)
[INFO] LMS7002M calibration: Volumes, Caching, Disable
[INFO] LMS7002M: LNA1, LNA2, LNA3, LNA4, LNA5, LNA6, LNA7, LNA8, LNA9, LNA10, LNA11, LNA12, LNA13, LNA14, LNA15, LNA16, LNA17, LNA18, LNA19, LNA20, LNA21, LNA22, LNA23, LNA24, LNA25, LNA26, LNA27, LNA28, LNA29, LNA30, LNA31, LNA32, LNA33, LNA34, LNA35, LNA36, LNA37, LNA38, LNA39, LNA40, LNA41, LNA42, LNA43, LNA44, LNA45, LNA46, LNA47, LNA48, LNA49, LNA50, LNA51, LNA52, LNA53, LNA54, LNA55, LNA56, LNA57, LNA58, LNA59, LNA60, LNA61, LNA62, LNA63, LNA64, LNA65, LNA66, LNA67, LNA68, LNA69, LNA70, LNA71, LNA72, LNA73, LNA74, LNA75, LNA76, LNA77, LNA78, LNA79, LNA80, LNA81, LNA82, LNA83, LNA84, LNA85, LNA86, LNA87, LNA88, LNA89, LNA90, LNA91, LNA92, LNA93, LNA94, LNA95, LNA96, LNA97, LNA98, LNA99, LNA100, LNA101, LNA102, LNA103, LNA104, LNA105, LNA106, LNA107, LNA108, LNA109, LNA110, LNA111, LNA112, LNA113, LNA114, LNA115, LNA116, LNA117, LNA118, LNA119, LNA120, LNA121, LNA122, LNA123, LNA124, LNA125, LNA126, LNA127, LNA128, LNA129, LNA130, LNA131, LNA132, LNA133, LNA134, LNA135, LNA136, LNA137, LNA138, LNA139, LNA140, LNA141, LNA142, LNA143, LNA144, LNA145, LNA146, LNA147, LNA148, LNA149, LNA150, LNA151, LNA152, LNA153, LNA154, LNA155, LNA156, LNA157, LNA158, LNA159, LNA160, LNA161, LNA162, LNA163, LNA164, LNA165, LNA166, LNA167, LNA168, LNA169, LNA170, LNA171, LNA172, LNA173, LNA174, LNA175, LNA176, LNA177, LNA178, LNA179, LNA180, LNA181, LNA182, LNA183, LNA184, LNA185, LNA186, LNA187, LNA188, LNA189, LNA190, LNA191, LNA192, LNA193, LNA194, LNA195, LNA196, LNA197, LNA198, LNA199, LNA200, LNA201, LNA202, LNA203, LNA204, LNA205, LNA206, LNA207, LNA208, LNA209, LNA210, LNA211, LNA212, LNA213, LNA214, LNA215, LNA216, LNA217, LNA218, LNA219, LNA220, LNA221, LNA222, LNA223, LNA224, LNA225, LNA226, LNA227, LNA228, LNA229, LNA230, LNA231, LNA232, LNA233, LNA234, LNA235, LNA236, LNA237, LNA238, LNA239, LNA240, LNA241, LNA242, LNA243, LNA244, LNA245, LNA246, LNA247, LNA248, LNA249, LNA250, LNA251, LNA252, LNA253, LNA254, LNA255, LNA256, LNA257, LNA258, LNA259, LNA260, LNA261, LNA262, LNA263, LNA264, LNA265, LNA266, LNA267, LNA268, LNA269, LNA270, LNA271, LNA272, LNA273, LNA274, LNA275, LNA276, LNA277, LNA278, LNA279, LNA280, LNA281, LNA282, LNA283, LNA284, LNA285, LNA286, LNA287, LNA288, LNA289, LNA290, LNA291, LNA292, LNA293, LNA294, LNA295, LNA296, LNA297, LNA298, LNA299, LNA300, LNA301, LNA302, LNA303, LNA304, LNA305, LNA306, LNA307, LNA308, LNA309, LNA310, LNA311, LNA312, LNA313, LNA314, LNA315, LNA316, LNA317, LNA318, LNA319, LNA320, LNA321, LNA322, LNA323, LNA324, LNA325, LNA326, LNA327, LNA328, LNA329, LNA330, LNA331, LNA332, LNA333, LNA334, LNA335, LNA336, LNA337, LNA338, LNA339, LNA340, LNA341, LNA342, LNA343, LNA344, LNA345, LNA346, LNA347, LNA348, LNA349, LNA350, LNA351, LNA352, LNA353, LNA354, LNA355, LNA356, LNA357, LNA358, LNA359, LNA360, LNA361, LNA362, LNA363, LNA364, LNA365, LNA366, LNA367, LNA368, LNA369, LNA370, LNA371, LNA372, LNA373, LNA374, LNA375, LNA376, LNA377, LNA378, LNA379, LNA380, LNA381, LNA382, LNA383, LNA384, LNA385, LNA386, LNA387, LNA388, LNA389, LNA390, LNA391, LNA392, LNA393, LNA394, LNA395, LNA396, LNA397, LNA398, LNA399, LNA400, LNA401, LNA402, LNA403, LNA404, LNA405, LNA406, LNA407, LNA408, LNA409, LNA410, LNA411, LNA412, LNA413, LNA414, LNA415, LNA416, LNA417, LNA418, LNA419, LNA420, LNA421, LNA422, LNA423, LNA424, LNA425, LNA426, LNA427, LNA428, LNA429, LNA430, LNA431, LNA432, LNA433, LNA434, LNA435, LNA436, LNA437, LNA438, LNA439, LNA440, LNA441, LNA442, LNA443, LNA444, LNA445, LNA446, LNA447, LNA448, LNA449, LNA450, LNA451, LNA452, LNA453, LNA454, LNA455, LNA456, LNA457, LNA458, LNA459, LNA460, LNA461, LNA462, LNA463, LNA464, LNA465, LNA466, LNA467, LNA468, LNA469, LNA470, LNA471, LNA472, LNA473, LNA474, LNA475, LNA476, LNA477, LNA478, LNA479, LNA480, LNA481, LNA482, LNA483, LNA484, LNA485, LNA486, LNA487, LNA488, LNA489, LNA490, LNA491, LNA492, LNA493, LNA494, LNA495, LNA496, LNA497, LNA498, LNA499, LNA500, LNA501, LNA502, LNA503, LNA504, LNA505, LNA506, LNA507, LNA508, LNA509, LNA510, LNA511, LNA512, LNA513, LNA514, LNA515, LNA516, LNA517, LNA518, LNA519, LNA520, LNA521, LNA522, LNA523, LNA524, LNA525, LNA526, LNA527, LNA528, LNA529, LNA530, LNA531, LNA532, LNA533, LNA534, LNA535, LNA536, LNA537, LNA538, LNA539, LNA540, LNA541, LNA542, LNA543, LNA544, LNA545, LNA546, LNA547, LNA548, LNA549, LNA550, LNA551, LNA552, LNA553, LNA554, LNA555, LNA556, LNA557, LNA558, LNA559, LNA560, LNA561, LNA562, LNA563, LNA564, LNA565, LNA566, LNA567, LNA568, LNA569, LNA570, LNA571, LNA572, LNA573, LNA574, LNA575, LNA576, LNA577, LNA578, LNA579, LNA580, LNA581, LNA582, LNA583, LNA584, LNA585, LNA586, LNA587, LNA588, LNA589, LNA590, LNA591, LNA592, LNA593, LNA594, LNA595, LNA596, LNA597, LNA598, LNA599, LNA600, LNA601, LNA602, LNA603, LNA604, LNA605, LNA606, LNA607, LNA608, LNA609, LNA610, LNA611, LNA612, LNA613, LNA614, LNA615, LNA616, LNA617, LNA618, LNA619, LNA620, LNA621, LNA622, LNA623, LNA624, LNA625, LNA626, LNA627, LNA628, LNA629, LNA630, LNA631, LNA632, LNA633, LNA634, LNA635, LNA636, LNA637, LNA638, LNA639, LNA640, LNA641, LNA642, LNA643, LNA644, LNA645, LNA646, LNA647, LNA648, LNA649, LNA650, LNA651, LNA652, LNA653, LNA654, LNA655, LNA656, LNA657, LNA658, LNA659, LNA660, LNA661, LNA662, LNA663, LNA664, LNA665, LNA666, LNA667, LNA668, LNA669, LNA670, LNA671, LNA672, LNA673, LNA674, LNA675, LNA676, LNA677, LNA678, LNA679, LNA680, LNA681, LNA682, LNA683, LNA684, LNA685, LNA686, LNA687, LNA688, LNA689, LNA690, LNA691, LNA692, LNA693, LNA694, LNA695, LNA696, LNA697, LNA698, LNA699, LNA700, LNA701, LNA702, LNA703, LNA704, LNA705, LNA706, LNA707, LNA708, LNA709, LNA710, LNA711, LNA712, LNA713, LNA714, LNA715, LNA716, LNA717, LNA718, LNA719, LNA720, LNA721, LNA722, LNA723, LNA724, LNA725, LNA726, LNA727, LNA728, LNA729, LNA730, LNA731, LNA732, LNA733, LNA734, LNA735, LNA736, LNA737, LNA738, LNA739, LNA740, LNA741, LNA742, LNA743, LNA744, LNA745, LNA746, LNA747, LNA748, LNA749, LNA750, LNA751, LNA752, LNA753, LNA754, LNA755, LNA756, LNA757, LNA758, LNA759, LNA760, LNA761, LNA762, LNA763, LNA764, LNA765, LNA766, LNA767, LNA768, LNA769, LNA770, LNA771, LNA772, LNA773, LNA774, LNA775, LNA776, LNA777, LNA778, LNA779, LNA780, LNA781, LNA782, LNA783, LNA784, LNA785, LNA786, LNA787, LNA788, LNA789, LNA790, LNA791, LNA792, LNA793, LNA794, LNA795, LNA796, LNA797, LNA798, LNA799, LNA800, LNA801, LNA802, LNA803, LNA804, LNA805, LNA806, LNA807, LNA808, LNA809, LNA810, LNA811, LNA812, LNA813, LNA814, LNA815, LNA816, LNA817, LNA818, LNA819, LNA820, LNA821, LNA822, LNA823, LNA824, LNA825, LNA826, LNA827, LNA828, LNA829, LNA830, LNA831, LNA832, LNA833, LNA834, LNA835, LNA836, LNA837, LNA838, LNA839, LNA840, LNA841, LNA842, LNA843, LNA844, LNA845, LNA846, LNA847, LNA848, LNA849, LNA850, LNA851, LNA852, LNA853, LNA854, LNA855, LNA856, LNA857, LNA858, LNA859, LNA860, LNA861, LNA862, LNA863, LNA864, LNA865, LNA866, LNA867, LNA868, LNA869, LNA870, LNA871, LNA872, LNA873, LNA874, LNA875, LNA876, LNA877, LNA878, LNA879, LNA880, LNA881, LNA882, LNA883, LNA884, LNA885, LNA886, LNA887, LNA888, LNA889, LNA890, LNA891, LNA892, LNA893, LNA894, LNA895, LNA896, LNA897, LNA898, LNA899, LNA900, LNA901, LNA902, LNA903, LNA904, LNA905, LNA906, LNA907, LNA908, LNA909, LNA910, LNA911, LNA912, LNA913, LNA914, LNA915, LNA916, LNA917, LNA918, LNA919, LNA920, LNA921, LNA922, LNA923, LNA924, LNA925, LNA926, LNA927, LNA928, LNA929, LNA930, LNA931, LNA932, LNA933, LNA934, LNA935, LNA936, LNA937, LNA938, LNA939, LNA940, LNA941, LNA942, LNA943, LNA944, LNA945, LNA946, LNA947, LNA948, LNA949, LNA950, LNA951, LNA952, LNA953, LNA954, LNA955, LNA956, LNA957, LNA958, LNA959, LNA960, LNA961, LNA962, LNA963, LNA964, LNA965, LNA966, LNA967, LNA968, LNA969, LNA970, LNA971, LNA972, LNA973, LNA974, LNA975, LNA976, LNA977, LNA978, LNA979, LNA980, LNA981, LNA982, LNA983, LNA984, LNA985, LNA986, LNA987, LNA988, LNA989, LNA990, LNA991, LNA992, LNA993, LNA994, LNA995, LNA996, LNA997, LNA998, LNA999, LNA1000, LNA1001, LNA1002, LNA1003, LNA1004, LNA1005, LNA1006, LNA1007, LNA1008, LNA1009, LNA1010, LNA1011, LNA1012, LNA1013, LNA1014, LNA1015, LNA1016, LNA1017, LNA1018, LNA1019, LNA1020, LNA1021, LNA1022, LNA1023, LNA1024, LNA1025, LNA1026, LNA1027, LNA1028, LNA1029, LNA1030, LNA1031, LNA1032, LNA1033, LNA1034, LNA1035, LNA1036, LNA1037, LNA1038, LNA1039, LNA1040, LNA1041, LNA1042, LNA1043, LNA1044, LNA1045, LNA1046, LNA1047, LNA1048, LNA1049, LNA1050, LNA1051, LNA1052, LNA1053, LNA1054, LNA1055, LNA1056, LNA1057, LNA1058, LNA1059, LNA1060, LNA1061, LNA1062, LNA1063, LNA1064, LNA1065, LNA1066, LNA1067, LNA1068, LNA1069, LNA1070, LNA1071, LNA1072, LNA1073, LNA1074, LNA1075, LNA1076, LNA1077, LNA1078, LNA1079, LNA1080, LNA1081, LNA1082, LNA1083, LNA1084, LNA1085, LNA1086, LNA1087, LNA1088, LNA1089, LNA1090, LNA1091, LNA1092, LNA1093, LNA1094, LNA1095, LNA1096, LNA1097, LNA1098, LNA1099, LNA1100, LNA1101, LNA1102, LNA1103, LNA1104, LNA1105, LNA1106, LNA1107, LNA1108, LNA1109, LNA1110, LNA1111, LNA1112, LNA1113, LNA1114, LNA1115, LNA1116, LNA1117, LNA1118, LNA1119, LNA1120, LNA1121, LNA1122, LNA1123, LNA1124, LNA1125, LNA1126, LNA1127, LNA1128, LNA1129, LNA1130, LNA1131, LNA1132, LNA1133, LNA1134, LNA1135, LNA1136, LNA1137, LNA1138, LNA1139, LNA1140, LNA1141, LNA1142, LNA1143, LNA1144, LNA1145, LNA1146, LNA1147, LNA1148, LNA1149, LNA1150, LNA1151, LNA1152, LNA1153, LNA1154, LNA1155, LNA1156, LNA1157, LNA1158, LNA1159, LNA1160, LNA1161, LNA1162, LNA1163, LNA1164, LNA1165, LNA1166, LNA1167, LNA1168, LNA1169, LNA1170, LNA1171, LNA1172, LNA1173, LNA1174, LNA1175, LNA1176, LNA1177, LNA1178, LNA1179, LNA1180, LNA1181, LNA1182, LNA1183, LNA1184, LNA1185, LNA1186, LNA1187, LNA1188, LNA1189, LNA1190, LNA1191, LNA1192, LNA1193, LNA1194, LNA1195, LNA1196, LNA1197, LNA1198, LNA1199, LNA1200, LNA1201, LNA1202, LNA1203, LNA1204, LNA1205, LNA1206, LNA1207, LNA1208, LNA1209, LNA1210, LNA1211, LNA1212, LNA1213, LNA1214, LNA1215, LNA1216, LNA1217, LNA1218, LNA1219, LNA1220, LNA1221, LNA1222, LNA1223, LNA1224, LNA1225, LNA1226, LNA1227, LNA1228, LNA1229, LNA1230, LNA1231, LNA1232, LNA1233, LNA1234, LNA1235, LNA1236, LNA1237, LNA1238, LNA1239, LNA1240, LNA1241, LNA1242, LNA1243, LNA1244, LNA1245, LNA1246, LNA1247, LNA1248, LNA1249, LNA1250, LNA1251, LNA1252, LNA1253, LNA1254, LNA1255, LNA1256, LNA1257, LNA1258, LNA1259, LNA1260, LNA1261, LNA1262, LNA1263, LNA1264, LNA1265, LNA1266, LNA1267, LNA1268, LNA1269, LNA1270, LNA1271, LNA1272, LNA1273, LNA1274, LNA1275, LNA1276, LNA1277, LNA1278, LNA1279, LNA1280, LNA1281, LNA1282, LNA1283, LNA1284, LNA1285, LNA1286, LNA1287, LNA1288, LNA1289, LNA1290, LNA1291, LNA1292, LNA1293, LNA1294, LNA1295, LNA1296, LNA1297, LNA1298, LNA1299, LNA1300, LNA1301, LNA1302, LNA1303, LNA1304, LNA1305, LNA1306, LNA1307, LNA1308, LNA1309, LNA1310, LNA1311, LNA1312, LNA1313, LNA1314, LNA1315, LNA1316, LNA1317, LNA1318, LNA1319, LNA1320, LNA1321, LNA1322, LNA1323, LNA1324, LNA1325, LNA1326, LNA1327, LNA1328, LNA1329, LNA1330, LNA1331, LNA1332, LNA1333, LNA1334, LNA1335, LNA1336, LNA1337, LNA1338, LNA1339, LNA1340, LNA1341, LNA1342, LNA1343, LNA1344, LNA1345, LNA1346, LNA1347, LNA1348, LNA1349, LNA1350, LNA1351, LNA1352, LNA1353, LNA1354, LNA1355, LNA1356, LNA1357, LNA1358, LNA1359, LNA1360, LNA1361, LNA1362, LNA1363, LNA1364, LNA1365, LNA1366, LNA1367, LNA1368, LNA1369, LNA1370, LNA1371, LNA1372, LNA1373, LNA1374, LNA1375, LNA1376, LNA1377, LNA1378, LNA1379, LNA1380, LNA1381, LNA1382, LNA1383, LNA1384, LNA1385, LNA1386, LNA1387, LNA1388, LNA1389, LNA1390, LNA1391, LNA1392, LNA1393, LNA1394, LNA1395, LNA1396, LNA1397, LNA1398, LNA1399, LNA1400, LNA1401, LNA1402, LNA1403, LNA1404, LNA1405, LNA1406, LNA1407, LNA1408, LNA1409, LNA1410, LNA1411, LNA1412, LNA1413, LNA1414, LNA1415, LNA1416, LNA1417, LNA1418, LNA1419, LNA1420, LNA1421, LNA1422, LNA1423, LNA1424, LNA1425, LNA1426, LNA1427, LNA1428, LNA1429, LNA1430, LNA1431, LNA1432, LNA1433, LNA1434, LNA1435, LNA1436, LNA1437, LNA1438, LNA1439, LNA1440, LNA1441, LNA1442, LNA1443, LNA1444, LNA1445, LNA1446, LNA1447, LNA1448, LNA1449, LNA1450, LNA1451, LNA1452, LNA1453, LNA1454, LNA1455, LNA1456, LNA1457, LNA1458, LNA1459, LNA1460, LNA1461, LNA1462, LNA1463, LNA1464, LNA1465, LNA1466, LNA1467, LNA1468, LNA1469, LNA1470, LNA1471, LNA1472, LNA1473, LNA1474, LNA1475, LNA1476, LNA1477, LNA1478, LNA1479, LNA1480, LNA1481, LNA1482, LNA1483, LNA1484, LNA1485, LNA1486, LNA1487, LNA1488, LNA1489, LNA1490, LNA1491, LNA1492, LNA1493, LNA1494, LNA1495, LNA1496, LNA1497, LNA1498, LNA1499, LNA1500, LNA1501, LNA1502, LNA1503, LNA1504, LNA1505, LNA1506, LNA1507, LNA1508, LNA1509, LNA1510, LNA1511, LNA1512, LNA1513, LNA1514, LNA1515, LNA1516, LNA1517, LNA1518, LNA1519, LNA1520, LNA1521, LNA1522, LNA1523, LNA1524, LNA1525, LNA1526, LNA1527, LNA1528, LNA1529, LNA1530, LNA1531, LNA1532, LNA1533, LNA1534, LNA1535, LNA1536, LNA1537, LNA1538, LNA1539, LNA1540, LNA1541, LNA1542, LNA1543, LNA1544, LNA1545, LNA1546, LNA1547, LNA1548, LNA1549, LNA1550, LNA1551, LNA1552, LNA1553, LNA1554, LNA1555, LNA1556, LNA1557, LNA1558, LNA1559, LNA1560, LNA1561, LNA1562, LNA1563, LNA1564, LNA1565, LNA1566, LNA1567, LNA1568, LNA1569, LNA1570, LNA1571, LNA1572, LNA1573, LNA1574, LNA1575, LNA1576, LNA1577, LNA1578, LNA1579, LNA1580, LNA1581, LNA1582, LNA1583, LNA1584, LNA1585, LNA1586, LNA1587, LNA1588, LNA1589, LNA1590, LNA1591, LNA1592, LNA1593, LNA1594, LNA1595, LNA1596, LNA1597, LNA1598, LNA1599, LNA1600, LNA1601, LNA1602, LNA1603, LNA1604, LNA1605, LNA1606, LNA1607, LNA1608, LNA1609, LNA1610, LNA1611, LNA1612, LNA1613, LNA1614, LNA1615, LNA1616, LNA1617, LNA1618, LNA1619, LNA1620, LNA1621, LNA1622, LNA1623, LNA1624, LNA1625, LNA1626, LNA1627, LNA1628, LNA1629, LNA1630, LNA1631, LNA1632, LNA1633, LNA1634, LNA1635, LNA1636, LNA1637, LNA1638, LNA1639, LNA1640, LNA1641, LNA1642, LNA1643, LNA1644, LNA1645, LNA1646, LNA1647, LNA1648, LNA1649, LNA1650, LNA1651, LNA1652, LNA1653, LNA1654, LNA1655, LNA1656, LNA1657, LNA1658, LNA1659, LNA1660, LNA1661, LNA1662, LNA1663, LNA1664, LNA1665, LNA1666, LNA1667, LNA1668, LNA1669, LNA1670, LNA1671, LNA1672, LNA1673, LNA1674, LNA1675, LNA1676, LNA1677, LNA1678, LNA1679, LNA1680, LNA1681, LNA1682, LNA1683, LNA1684, LNA1685, LNA1686, LNA1687, LNA1688, LNA1689, LNA1690, LNA1691, LNA1692, LNA1693, LNA1694, LNA1695, LNA1696, LNA1697, LNA1698, LNA1699, LNA1700, LNA1701, LNA1702, LNA1703, LNA1704, LNA1705, LNA1706, LNA1707, LNA1708, LNA1709, LNA1710, LNA1711, LNA1712, LNA1713, LNA1714, LNA1715, LNA1716, LNA1717, LNA1718, LNA1719, LNA1720, LNA1721, LNA1722, LNA1723, LNA1724, LNA1725, LNA1726, LNA1727, LNA1728, LNA1729, LNA1730, LNA1731, LNA1732, LNA1733, LNA1734, LNA1735, LNA1736, LNA1737, LNA1738, LNA1739, LNA1740, LNA1741, LNA1742, LNA1743, LNA1744, LNA1745, LNA1746, LNA1747, LNA1748, LNA1749, LNA1750, LNA1751, LNA1752, LNA1753, LNA1754, LNA1755, LNA1756, LNA1757, LNA1758, LNA1759, LNA1760, LNA1761, LNA1762, LNA1763, LNA1764, LNA1765, LNA1766, LNA1767, LNA1768, LNA1769, LNA1770, LNA1771, LNA1772, LNA1773, LNA1774, LNA1775, L
```

```

*\param device a pointer to a device instance
*\param direction the channel direction RX or TX
*\param channel an available channel on the device
*\param value the new amplification value in dB
*\return an error code or 0 for success
*/

```

```

SOAPY_SDR_API int SoapySDRDevice_setGain(SoapySDRDevice *device, const int direction, const size_t
channel, const double value);

```

通过上述信息，我们可以这样调用API来设置增益：

```
int success = SoapySDRDevice_setGain(sdr, SOAPY_SDR_RX, 0, 20)
```

这样接收通道就会有20dB的增益。就是这么简单，整个API的文档都是这么写的，所以要加入其它函数也是小事一桩。

先整理一下

我们现在要实现一个新的工具。我们要从小程序起步，比如实现一个频谱检测程序。我们可以对于每个频率读取1024个采样点，并计算RMS值。我们可以设置开始频率和结束频率，并且设置每次读取间隔多少Hz。

首先我们知道：程序代码对于硬件工程师来讲并不熟悉，我们无法阅读大量代码，所以最好把这个例子分解开来，这样我们的主循环可以小一点，维护代码也方便。我们需要用到函数和指针，所以你最好先学习一点C语言基础（我们自己也需要做点练习）。

我们会写一些函数（这些可以看作是原型）：

```

struct SoapySDRDevice *Setup(void);
void DeviceInfo();
void Read_1024_samples(struct SoapySDRDevice *sdr, double freq, complex float *buffer);
void Close();

```

SoapySDRDevice

这个函数相对复杂，它会返回一个SoapySDRDevice结构体，并且把一个指针指向这个创建的SDR设备。这个指针是其它所有函数要用的，它们通过这种方式与LimeSDR交互。

这个函数可以像这样调用：

```
struct SoapySDRDevice *sdr = Setup();
```

```

struct SoapySDRDevice *Setup(void)
{
    //enumerate devices
    SoapySDRKwargs *results = SoapySDRDevice_enumerate(NULL, &length);
    for (size_t i = 0; i < length; i++)
    {
        printf("Found device #d: ", (int)i);
        for (size_t j = 0; j < results.size; j++)
        {
            printf("%s=%s, ", results.keys[j], results.vals[j]);
        }
        printf("\n");
    }
    SoapySDRKwargsList_clear(results, length);
    //create device instance
    //args can be user defined or from the enumeration result
    SoapySDRKwargs args = {};
    SoapySDRKwargs_set(&args, "driver", "lime");
    SoapySDRDevice *sdr = SoapySDRDevice_make(&args);
    SoapySDRKwargs_clear(&args);
    if(sdr == NULL)
    {
        printf("SoapySDRDevice_make fail: %s\n", SoapySDRDevice_lastError());
        //return EXIT_FAILURE;
    }
    return sdr;
}

```

DeviceInfo

这是一个简单的函数，返回信息数据，在调用时需要传入SDR设备：

```
DeviceInfo(sdr);
```

```

void DeviceInfo(struct SoapySDRDevice *sdr)
{
    //query device info
    char** names = SoapySDRDevice_listAntennas(sdr, SOAPY_SDR_RX, 0, &length);
    printf("Rx antennas: ");
    for (size_t i = 0; i < length; i++) printf("%s, ", names);
    printf("\n");
    SoapySDRStrings_clear(&names, length);
    names = SoapySDRDevice_listGains(sdr, SOAPY_SDR_RX, 0, &length);
    printf("Rx gains: ");
    for (size_t i = 0; i < length; i++) printf("%s, ", names);
    printf("\n");
    SoapySDRStrings_clear(&names, length);
}

```

```
SoapySDRRange *ranges = SoapySDRDevice_getFrequencyRange(sdr, SOAPY_SDR_RX, 0, &length);
printf("Rx freq ranges: ");
for(size_t i = 0; i < length; i++) printf("[%g Hz -> %g Hz], ", ranges.minimum, ranges.maximum);
printf("\n");
free(ranges);
}
```

回复

举报

jamesshao8

楼主 | 发表于 2018-12-22 11:47:30 | 只看该作者

板凳



10	23	80
主题	帖子	积分

注册会员



积分 80

发消息

Read_1024_samples

一开始这个函数看上去有点可怕。你可以像这样调用它：

```
Read_1024_samples(sdr, frequency, buffn);
```

sdr是Setup()函数创建的，frequency是要调谐到的频率值，最后的buffn是用来接收读取的采样点的buffer。这些值会存储到buffn变量的指针中。

我们还在这个函数中加入了一些错误检查机制。这样可以避免调谐到支持的范围外。除此之外，我们加入了增益，并且强制设置LNAL作为接收天线。

```
void Read_1024_samples(struct SoapySDRDevice *sdr, double freq, complex float *buffer)
{
    // check freq is within range
    if (freq <10e3)
    {
        freq = 10e3;
        printf("Frequency out of range setting min: \n");
    }
    if (freq >3.8e9)
    {
        freq = 3.8e9;
        printf("Frequency out of range setting max: \n");
    }
    //apply settings
    if (SoapySDRDevice_setSampleRate(sdr, SOAPY_SDR_RX, 0, 1e6) != 0)
    {
        printf("setSampleRate fail: %s\n", SoapySDRDevice_lastError());
    }
    if (SoapySDRDevice_setAntenna(sdr, SOAPY_SDR_RX, 0, "LNAL") != 0)
    {
        printf("setAntenna fail: %s\n", SoapySDRDevice_lastError());
    }
    if (SoapySDRDevice_setGain(sdr, SOAPY_SDR_RX, 0, 20) != 0)
    {
        printf("setAntenna fail: %s\n", SoapySDRDevice_lastError());
    }
    if (SoapySDRDevice_setFrequency(sdr, SOAPY_SDR_RX, 0, freq, NULL) != 0)
    {
        printf("setFrequency fail: %s\n", SoapySDRDevice_lastError());
    }
    //setup a stream (complex floats)
    SoapySDRStream *rxStream;
    if (SoapySDRDevice_setupStream(sdr, &rxStream, SOAPY_SDR_RX, SOAPY_SDR_CF32, NULL, 0, NULL) != 0)
    {
        printf("setupStream fail: %s\n", SoapySDRDevice_lastError());
    }
    SoapySDRDevice_activateStream(sdr, rxStream, 0, 0, 0); //start streaming
    void *buffs[] = {buffer}; //array of buffers
    int flags; //flags set by receive operation
    long long timeNs; //timestamp for receive buffer
    int ret = SoapySDRDevice_readStream(sdr, rxStream, buffs, 1024, &flags, &timeNs, 100000);
    //shutdown the stream
    SoapySDRDevice_deactivateStream(sdr, rxStream, 0, 0); //stop streaming
    SoapySDRDevice_closeStream(sdr, rxStream);
}
```

Close

最终我们需要关闭之前用Setup函数打开的SDR设备，我们只需要调用：

```
close(sdr)
```

```
void Close(struct SoapySDRDevice *sdr2)
{
    //cleanup device handle
    SoapySDRDevice_unmake(sdr2);
    printf("Done\n");
}
```

```
//return EXIT_SUCCESS;  
}
```

最小的程序

我们的程序大概已经写了90%了，我们现在写一个最小型的例子。我们需要include下面这些文件：

```
#include <SoapySDR/Device.h>  
#include <SoapySDR/Formats.h>  
#include <stdio.h> //printf  
#include <stdlib.h> //free  
#include <complex.h>
```

这是唯一的全局变量

```
size_t length;
```

主函数只有这三行

```
struct SoapySDRDevice *sdr = Setup();  
DeviceInfo(sdr);  
Close(sdr);
```

一个频率监控工具

接下来只需要加入一小部分代码就可以开始创建很多应用程序了。这就是我们的频率监控例子：

我们在命令行传入3个参数：开始频率、结束频率、间隔。这个代码就会设置好LimeSDR，计算需要做多少次采集。根据这个结果，程序会在每个频率上读取1024个采样点，然后计算RMS值。RMS均方根，就是把所有值取平方，再求平均，再求平方根。这个结果会打印到终端里。注意sqrt命令要加入-lm才可以编译正常，可能你的电脑不一定要加。

```
int main(int argc, char *argv[])  
{  
    int argvcount=0;  
    if(argc!=4)  
    {  
        printf("%d",argc);  
        printf("invalid input");  
        exit(EXIT_FAILURE);  
    }  
    struct SoapySDRDevice *sdr = Setup();  
    DeviceInfo(sdr);  
    atexit(Cleanup);  
    // HERE WE CAN DO THINGS  
    // Lets do a site survey measuring every xxx Hz and displaying the power  
    double StartFreq = strtod(argv[1],NULL);  
    double EndFreq = strtod(argv[2],NULL);  
    double Interval =strtod(argv[3],NULL);  
    double NumberOfScanPoints= (EndFreq-StartFreq)/Interval;  
    printf("Starting RMS scan, Start is: %f, End is: %f, Interval is: %f\n",StartFreq,EndFreq,Interval);  
    printf("This gives a total of: %f Sample points",NumberOfScanPoints);  
    while (NumberOfScanPoints !=0)  
    {  
        Get_Samples(sdr);  
        //create a re-usable buffer for rx samples  
        complex float buffn[1024];  
        double frequency = StartFreq + (Interval * ((EndFreq/Interval)-NumberOfScanPoints));  
        Read_1024_samples(sdr, frequency,buffn);  
        //printf("ret=%d, flags=%d, timeNs=%lld\n", ret, flags, timeNs);  
        float realavg = 0;  
        for (int x = 0; x < 1024; x++)  
        {  
            printf("%f + i%f\n", creal(buffn[x]), cimag(buffn[x]));  
            realavg+= (creal(buffn[x])*creal(buffn[x]));  
        }  
        realavg = sqrt(realavg/1024);  
        printf("Freq is: %f, RMS is: %f \n",frequency,realavg);  
        NumberOfScanPoints--;  
    }  
    Close(sdr);  
}
```

编译完成并运行程序，输出如下：


```
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$ gcc -std=c99 soapy_api.c -lSoapySDR -lm -lsoapy_api.bin
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$ ./soapy_api.bin 1e6 2e6 100e3
Found device #0: addr=1d50:6108, driver=lime, label=LimeSDR-USB [USB 2.0] 9060B00472118, media=USB 2.0, module=STREAM, name=LimeSDR-USB, serial=0009060B00472118
,
[INFO] Make connection: 'LimeSDR-USB [USB 2.0] 9060B00472118'
[INFO] Reference clock 30.720 MHz
[INFO] Device name: LimeSDR-USB
[INFO] Reference: 30.72 MHz
[INFO] Init LMS7002M(0)
[INFO] Ver=7, Rev=1, Mask=1
[INFO] LMS7002M calibration values caching Disable
Rx antennas: NONE, LNAH, LNAL, LNAW, LB1, LB2,
Rx gains: LNA, TIA, PGA,
Rx freq ranges: [0 Hz -> 3.8e+09 Hz],
Starting RMS scan, Start is: 1000000.000000, End is: 2000000.000000, Interval is
: 100000.000000
This gives a total of: 10.000000 Sample pointsFreq is: 2000000.000000, RMS is: 0
.002227
Freq is: 2100000.000000, RMS is: 0.001122
Freq is: 2200000.000000, RMS is: 0.000579
Freq is: 2300000.000000, RMS is: 0.002567
Freq is: 2400000.000000, RMS is: 0.001817
Freq is: 2500000.000000, RMS is: 0.000418
Freq is: 2600000.000000, RMS is: 0.002236
Freq is: 2700000.000000, RMS is: 0.000648
Freq is: 2800000.000000, RMS is: 0.001009
Freq is: 2900000.000000, RMS is: 0.001249
Done
karl@bronte:~/Documents/LimeSDR/Soapy_API_Work$
```

在命令参数里加入LNA选择和增益设置应该也很简单。另外，把输出保存到文件中，在Linux下也很好实现，只需要这样运行：./soapy_api.bin 1e6 2e6 100e3 >output.txt

最后

C语言很有用，要实现更强大的程序也美问题。解码、编码、发射、接收都是可以实现的，就像上面这个例子一样简单。

我们希望从这个简单的SoapySDR API里会给予你一些启发，使用这个方式，我们就不受限于Pothos和GNU Radio提供的模块了，我们可以创建全新的程序，你也可以选用熟悉的语言（C、C++、Python都可以）。

[回复](#)

[举报](#)

[发新帖](#)

[返回列表](#)