```
/*
 * Trabalho de Graduação II - Controle antecipativo de sistemas de conforto térmico usando RFID
 *
 * Autor: André Abreu Rodrigues de Almeida
 *        Graduando em Engenharia Mecatrônica
 *        Matrícula: 12/0007100
 * Departamento de Engenharia Elétrica
 * Universidade de Brasília - UnB
 *
 *
 * Brasília, Agosto a Dezembro de 2019
 *
 */

/* Aviso: A biblioteca 'Octane SDK' foi desenvolvida e é propriedade da empresa IMPINJ, INC., e está sendo utilizada com base na licença de software aberto descrita a seguir
 *
 * Para mais informações, visitar o site: https://support.impinj.com/hc/en-us/articles/360000468370-Software-Tools-License-Disclaimer
 */


/*PLEASE READ THE FOLLOWING LICENSE & DISCLAIMER (“AGREEMENT”) CAREFULLY BEFORE USING ANY SOFTWARE TOOLS (AS DEFINED BELOW) MADE AVAILABLE TO YOU (“LICENSEE”) BY IMPINJ, INC. (“IMPINJ”). BY USING THE SOFTWARE TOOLS, YOU ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD ALL THE TERMS AND CONDITIONS OF THE AGREEMENT, YOU WILL BE CONSENTING TO BE BOUND BY THEM, AND YOU ARE AUTHORIZED TO DO SO. IF YOU DO NOT ACCEPT THESE TERMS AND CONDITIONS, DO NOT USE THE SOFTWARE TOOLS.

1. PURPOSE OF AGREEMENT. From time to time, Impinj technical personnel may make available to Licensee certain software, including code (in source and object form), tools, libraries, configuration files, translations, and related documentation (collectively, “Software Tools”), upon specific request or to assist with a specific deployment. This Agreement sets forth Licensee's limited rights and Impinj's limited obligations with respect to the Software Tools. Licensee acknowledges that Impinj provides the Software Tools free of charge. This Agreement does not grant any rights with respect to Impinj standalone software products (e.g., ItemSense, ItemEncode, SpeedwayConnect) or the firmware on Impinj hardware, all of which are subject to separate license terms.

2. LIMITED LICENSE. Subject to the terms and conditions of this Agreement, Impinj hereby grants to Licensee a limited, royalty-free, worldwide, non-exclusive, perpetual and irrevocable (except as set forth below), non-transferable license, without right of sublicense, to (a) use the Software Tools and (b) only with respect to Software Tools provided in source code form, modify and create derivative works of such Software Tools, in each case, solely for Licensee’s internal development related to the deployment of Impinj products (“Purpose”). The Software Tools may only be used by employees of Licensee that must have access to the Software Tools in connection with the Purpose.

3. TERMINATION. Impinj may immediately terminate this Agreement if Licensee
```

```
      breaches any provision hereof. Upon the termination of this Agreement,
      Licensee must (a) discontinue all use of the Software Tools, (b) uninstall
      the Software Tools from its systems, (c) destroy or return to Impinj all
      copies of the Software Tools and any other materials provided by Impinj, and
       (d) promptly provide Impinj with written confirmation (including via email)
       of Licensee's compliance with these provisions. Sections 4-10 will survive
      termination of this Agreement.
28
29    4. OWNERSHIP. The Software Tools are licensed, not sold, by Impinj to
      Licensee. Impinj and its suppliers own and retain all right, title, and
      interest, including all intellectual property rights, in and to the Software
       Tools. Except for those rights expressly granted in this Agreement, no
      other rights are granted, either express or implied, to Licensee. Impinj
      reserves the right to develop, price and sell software products that have
      features similar to or competitive with Software Tools. Licensee grants
      Impinj a limited, royalty-free, worldwide, perpetual and irrevocable,
      transferable, sublicensable, license to Licensee's derivative works of
      Software Tools; provided that Licensee has no obligation under this
      Agreement to deliver to Impinj any such derivative works.
30
31    5. CONFIDENTIALITY. In order to protect the trade secrets and proprietary
      know-how contained in the Software Tools, Licensee will not decompile,
      disassemble, or reverse engineer, or otherwise attempt to gain access to the
       source code or algorithms of the Software Tools (unless Impinj provides the
       Software Tools in source code format). Licensee will maintain the
      confidentiality of and not disclose to any third party: (a) all non-public
      information disclosed by Impinj to Licensee under this Agreement and (b) all
       performance data and all other information obtained through the Software
      Tools.
32
33    6. WARRANTY DISCLAIMER. LICENSEE AC KNOWLEDGES THAT IMPINJ PROVIDES THE
      SOFTWARE TOOLS FREE OF CHARGE AND ONLY FOR THE PURPOSE. ACCORDINGLY, THE
      SOFTWARE TOOLS ARE PROVIDED "AS IS" WITHOUT QUALITY CHECK, AND IMPINJ DOES
      NOT WARRANT THAT THE SOFTWARE TOOLS WILL OPERATE WITHOUT ERROR OR
      INTERRUPTION OR MEET ANY PERFORMANCE STANDARD OR OTHER EXPECTATION. IMPINJ
      EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING THE
      IMPLIED WARRANTIES OF MERCHANTABILITY, NONINFRINGEMENT, QUALITY, ACCURACY,
      AND FITNESS FOR A PARTICULAR PURPOSE. IMPINJ IS NOT OBLIGATED IN ANY WAY TO
      PROVIDE SUPPORT OR OTHER MAINTENANCE WITH RESPECT TO THE SOFTWARE TOOLS.
34
35    7. LIMITATION OF LIABILITY. THE TOTAL LIABILITY OF IMPINJ ARISING OUT OF OR
      RELATED TO THE SOFTWARE TOOLS WILL NOT EXCEED THE TOTAL AMOUNT PAID BY
      LICENSEE TO IMPINJ PURSUANT TO THIS AGREEMENT. IN NO EVENT WILL IMPINJ HAVE
      LIABILITY FOR ANY INDIRECT, INCIDENTAL, SPECIAL, OR CONSEQUENTIAL DAMAGES,
      EVEN IF ADVISED OF THE POSSIBILITY OF THESE DAMAGES. THESE LIMITATIONS WILL
      APPLY NOTWITHSTANDING ANY FAILURE OF ESSENTIAL PURPOSE OF ANY LIMITED REMEDY
       IN THIS AGREEMENT.
36
37    8. THIRD PARTY SOFTWARE. The Software Tools may contain software created by a
      third party. Licensee's use of any such third party software is subject to
      the applicable license terms and this Agreement does not alter those license
       terms. Licensee may not subject any portion of the Software Tools to an
      open source license.
38
39    9. RESTRICTED USE. Licensee will comply with all applicable laws and
      regulations to preclude the acquisition by any governmental agency of
```

```
      unlimited rights to technical data, software, and documentation provided
      with Software Tools, and include the appropriate “Restricted Rights” or
      “Limited Rights” notices required by the applicable U.S. or foreign
      government agencies. Licensee will comply in all respects with all U.S. and
      foreign export and re-export laws and regulations applicable to the
      technology and documentation provided hereunder.
40
41    10. MISCELLANEOUS. This Agreement will be governed by the laws of the State of
       Washington, U.S.A without reference to conflict of law principles. All
      disputes arising out of or related to it, will be subject to the exclusive
      jurisdiction of the state and federal courts located in King County,
      Washington, and the parties agree and submit to the personal and exclusive
      jurisdiction and venue of these courts. Licensee will not assign this
      Agreement, directly or indirectly, by operation of law or otherwise, without
       the prior written consent of Impinj. This Agreement (and any applicable
      nondisclosure agreement) is the entire agreement between the parties
      relating to the Software Tools. No waiver or modification of this Agreement
      will be valid unless contained in a writing signed by each party.
42    */
43
44    #define DEBUG
45
46    using System;
47    using Impinj.OctaneSdk;
48    using System.Collections;
49    using System.Collections.Generic;
50    using System.IO;
51    using System.Text;
52    using System.Threading;
53
54    namespace TG2_RFID
55    {
56        class GlobalData
57        {
58            public static int RSSILowPassFilter = -60;
59
60            public static FileHandler filehandler = new FileHandler();
61        }
62
63        class Program
64        {
65            // Create a collection to hold all the ImpinjReader instances.
66            protected static List<ImpinjReader> readers = new List<ImpinjReader>
                ();
67
68            static void Main(/*string[] args*/)
69            {
70                //Console.WriteLine("Test Step");
71                //Curve cc = new Curve();
72                //Curve.PopulateCurveTest(cc);
73                //cc.PrintCurveInConsole();
74                //Console.WriteLine("MeanY:{0}", cc.CalculateMeanY());
75                //Console.WriteLine("MedianY:{0}", cc.GetMedianY());
76                //var maxPoint = cc.GetCurveMaxPoint();
77                //var minPoint = cc.GetCurveMinPoint();
78                //Console.WriteLine("MaxPoint<{0}, {1}>", maxPoint.Item1,
```

```
                maxPoint.Item2);
79              //Console.WriteLine("MinPoint<{0}, {1}>", minPoint.Item1,
                minPoint.Item2);
80              //Console.WriteLine("MinX:{0}, MaxX:{1}", cc.GetCurveMinX(),
                cc.GetCurveMaxX());
81              //var crossingPoint = cc.CalculateCrossingPoint();
82              //Console.WriteLine("CrossingPoint<{0}, {1}>",
                crossingPoint.Item1, crossingPoint.Item2);
83              //var peaks = cc.CalculatePeaks();
84              //Console.WriteLine("NPeaks: {0}", peaks.Count);
85              //foreach (var peak in peaks)
86              //{
87              //    Console.WriteLine("Peak: <{0},{1}>", peak.Item1,
                peak.Item2);
88              //}
89
90
91              //Console.WriteLine("End Test Step");
92              //return;
93
94              try
95              {
96                  // Sets the output file path
97                  GlobalData.filehandler.SetFileHandler();
98                  GlobalData.filehandler.CreateFile();
99
100                 // holds the paths to the readers
101                 string hostname1 = "speedwayr-10-9f-3f.local";
102                 string hostname2 = "speedwayr-10-9f-c8.local";
103                 string hostname3 = "speedwayr-10-9f-bb.local";
104
105                 // Create two reader instances and add them to the List of
                    readers.
106                 readers.Add(new ImpinjReader(hostname1, "Reader #1"));
107                 readers.Add(new ImpinjReader(hostname2, "Reader #2"));
108                 readers.Add(new ImpinjReader(hostname3, "Reader #3"));
109
110                 //Create map of rooms
111                 Project.RegisterNewAmbient(0, new Ambient("Area_Externa(0)"));
112                 Project.RegisterNewAmbient(1, new Ambient("Sala_Principal
                    (1)"));
113                 Project.RegisterNewAmbient(2, new Ambient("Sala_Reuniao(2)"));
114                 Project.RegisterNewAmbient(3, new Ambient("Corredor_Baias
                    (3)"));
115
116                 //Create map of transitions
117                 Transition transition1 = new Transition
                      (Project.GetAmbientInstance(0), "Reader #1", 1,
                      Project.GetAmbientInstance(1), "Reader #1", 2);
118                 Transition transition2 = new Transition
                      (Project.GetAmbientInstance(1), "Reader #2", 2,
                      Project.GetAmbientInstance(2), "Reader #2", 1);
119                 Transition transition3 = new Transition
                      (Project.GetAmbientInstance(1), "Reader #3", 1,
                      Project.GetAmbientInstance(3), "Reader #3", 2);
120                 Project.RegisterNewTransition(Tuple.Create<string, ushort>
```

```
                    ("Reader #1", 1), transition1);
121                 Project.RegisterNewTransition(Tuple.Create<string, ushort>
                    ("Reader #1", 2), transition1);
122                 Project.RegisterNewTransition(Tuple.Create<string, ushort>
                    ("Reader #2", 1), transition2);
123                 Project.RegisterNewTransition(Tuple.Create<string, ushort>
                    ("Reader #2", 2), transition2);
124                 Project.RegisterNewTransition(Tuple.Create<string, ushort>
                    ("Reader #3", 1), transition3);
125                 Project.RegisterNewTransition(Tuple.Create<string, ushort>
                    ("Reader #3", 2), transition3);
126
127                 //Create Map of Cardholders
128                 Project.PopulateProjectCardholders();
129
130                 // Loop through the List of readers to configure and start
                    them.
131                 foreach (ImpinjReader reader in readers)
132                 {
133                     // Connect to the reader
134                     reader.Connect();
135
136                     // Get the default settings
137                     // We'll use these as a starting point
138                     // and then modify the settings we're
139                     // interested in.
140                     Settings settings = reader.QueryDefaultSettings();
141
142                     settings.Report.IncludeAntennaPortNumber = true;
143                     settings.Report.IncludeFirstSeenTime = true;
144                     settings.Report.IncludeLastSeenTime = true;
145                     settings.Report.IncludeSeenCount = true;
146                     settings.Report.IncludeDopplerFrequency = true;
147                     settings.Report.IncludePeakRssi = true;
148                     // Send a tag report for every tag read
149                     settings.Report.Mode = ReportMode.Individual;//
                    BatchAfterStop;
150
151                     // Reading tags for 1 seconds every 0.25 second
152                     //settings.AutoStart.Mode = AutoStartMode.Periodic;
153                     //settings.AutoStart.PeriodInMs = 500;
154                     //settings.AutoStop.Mode = AutoStopMode.Duration;
155                     //settings.AutoStop.DurationInMs = 500;
156
157                     //Settings de antena
158                     settings.Antennas.DisableAll();
159                     settings.Antennas.GetAntenna(1).IsEnabled = true;
160                     settings.Antennas.GetAntenna(2).IsEnabled = true;
161                     // Set all the antennas to the max transmit power and
                    receive sensitivity
162                     settings.Antennas.TxPowerMax = true;
163                     settings.Antennas.RxSensitivityMax = true;
164                     // Or set all antennas to a specific value in dBm
165                     //settings.Antennas.TxPowerInDbm = 28.0;
166                     //settings.Antennas.RxSensitivityInDbm = -70.0;
167                     // Or set each antenna individually
```

```
168                     //settings.Antennas.GetAntenna(1).MaxTxPower = true;
169                     //settings.Antennas.GetAntenna(1).MaxRxSensitivity = true;
170                     //settings.Antennas.GetAntenna(2).TxPowerInDbm = 30.0;
171                     //settings.Antennas.GetAntenna(2).RxSensitivityInDbm =
                        -70.0;
172                     // ...
173
174
175                     settings.ReaderMode = ReaderMode.DenseReaderM8;
176
177
178
179                     // Apply the newly modified settings.
180                     reader.ApplySettings(settings);
181
182                     // Assign the TagsReported event handler.
183                     // This specifies which method to call
184                     // when tags reports are available.
185                     reader.TagsReported += Captura_tags;
186
187                     // Start reading.
188                     reader.Start();
189                 }
190
191                 // Wait for the user to press enter.
192                 //Console.WriteLine("Press enter to exit.");
193                 //Console.ReadKey();
194                 while(true)
195                 {
196                     Console.WriteLine("Press 0, 1, 2, and 3 to set debuggin
                        cardholder ambient and anything else to exit.");
197                     var stringConsole = Console.ReadKey().KeyChar;
198                     var aux = Convert.ToInt32(stringConsole) - 48;
199                     if (Project.realAmbient <= 3 && Project.realAmbient >= 0)
200                     {
201                         Project.realAmbient = (ushort)(aux);
202                     }
203                     else
204                     {
205                         Project.realAmbient = 0;
206                         break;
207                     }
208                 }
209
210                 // Stop all the readers and disconnect from them.
211                 foreach (ImpinjReader reader in readers)
212                 {
213                     try
214                     {
215                         reader.Stop();
216                         reader.Disconnect();
217                     }
218                     catch (OctaneSdkException) { }
219                 }
220             }
221             catch (OctaneSdkException e)
```

```
222             {
223                 // Handle Octane SDK errors.
224                 Console.WriteLine("Octane SDK exception: {0}", e.Message);
225             }
226             catch (Exception e)
227             {
228                 // Handle other .NET errors.
229                 Console.WriteLine("Exception : {0}", e.Message);
230             }
231
232
233             // Wait for the user to press enter.
234             Console.WriteLine("Press enter to exit.");
235             Console.ReadKey();
236         }
237
238
239
240         private static void Captura_tags(ImpinjReader sender, TagReport
              report)
241         {
242             foreach (Tag tag in report)
243             {
244                 if (Project.IsTagRegistered(tag) && tag.PeakRssiInDbm >
                      GlobalData.RSSILowPassFilter && Math.Abs
                      (tag.RfDopplerFrequency) > 0.5 )
245                 {
246                     Project.ReadingCardholderTag(tag, sender.Name);
247                     Project.ProcessCardholderData(tag, sender.Name);
248                     var individuo = Project.GetCardholder(tag.Epc.ToString());
249
250                     // Writes data to file
251                     GlobalData.filehandler.WriteToFile(individuo,
                        tag.Epc.ToString(), sender.Name, tag.AntennaPortNumber);
252
253                     // Debug
254                     // Console.WriteLine("RSSI: {0}, Doppler: {1}, 0:{2}, 2:
                        {3}, 3:{4}, 4:{5}, 5:{6}, 6:{7}", tag.PeakRssiInDbm,
                        tag.RfDopplerFrequency, individuo.GetAmbient(0).GetName(),
                         individuo.GetAmbient(2).GetName(), individuo.GetAmbient
                        (3).GetName(), individuo.GetAmbient(4).GetName(),
                        individuo.GetAmbient(5).GetName(), individuo.GetAmbient
                        (6).GetName());
255                     // Debug.end
256                     Console.WriteLine("Name: {0},    Peaks:{1},     Dopppler:
                        {2},    Combined:{3}", individuo.GetName(),
                        individuo.GetAmbient(0).GetName(), individuo.GetAmbient
                        (5).GetName(), individuo.GetAmbient(6).GetName());
257
258                 }
259             }
260         }
261
262
263
264
```

```
265
266         }
267     }
```