Report Iot Interfaces From Hardware To Software And Wireless Communication

Subject: TP Capteurs et environment, 2021

Author: Duc Dinh Anh
Email: ducda.m20ict@st.usth.edu.vn
Mobile: (+84) 982 086 651

July 2021

1 Compute geographic bounding box

1.1 *What is geographic bounding box ?

- **Bounding box** is the limitation or the border which contains all the sensor of our system, according to my knowledge bounding box is also the perfect area in which the sensor can interact efficiently. Beside, bounding box can determine the distance of the sensors and how far the signal can reach.

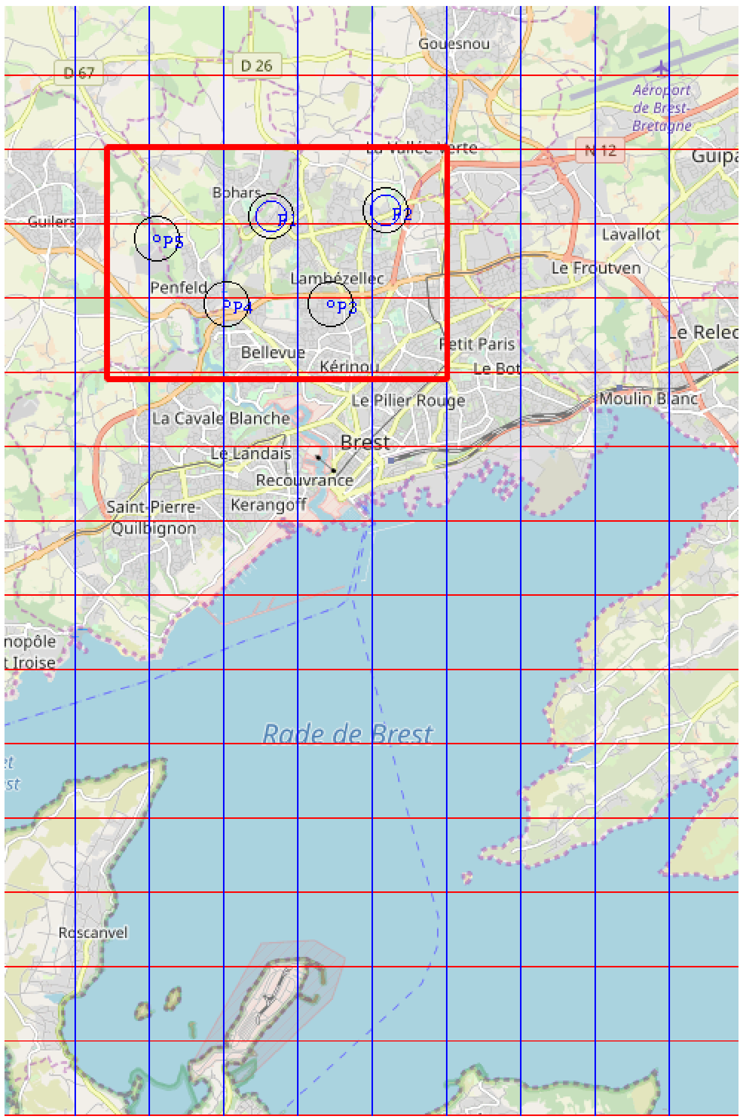

Figure 1: The red bounder is the bounding box in physical world.

1.2 *How can we determine the bounding box ?

- **Step 1**: We need to create the behaviour file whose name is "nodes-test-include.occ". In this file, we need to add two classes one is "Node" and the other is "Mux". "Node" represents the sensor, "Mux" is the method that displays all the output from "Node".

- **Step 2**: We need to get the x-y coordinator of each sensor by using the "Node" class in "nodes-test-include.occ" file. To do that we can access to the global array "NetLocation" in "halongtpnet.occ" file which contains all the information of each sensor.

- **Step 3**: After we find the bounding box, we need to use cooperation techniques to send the coordinator of bounding box to all sensor.

- ***Note**: To find the bounding box we need to find the most left and most right value of the coordinator.

1.3 Code

```
PROC Node([]CHAN OF diam.proto in,out, VAL INT Identity,
CHAN OF BYTE toMux)
  INT MyXLoc:
  INT MyYLoc:
  [MaxFanOut] BBox BBuf:
  BBox myBBox:
  SEQ
    -- MyXLoc, MyYLoc are variables which store
    MyXLoc := NetLocation[Identity][xLoc]
    MyYLoc := NetLocation[Identity][yLoc]
    -- Initial BBOX
    myBBox[Left] := MyXLoc
    myBBox[Right] := MyXLoc
    myBBox[Top] := MyYLoc
    myBBox[Bottom] := MyYLoc
```

```
    SEQ Turns=0 FOR MaxNodes-1
      SEQ
      -- Create input and ouput section
        PAR
          PAR i=0 FOR SIZE out
            out[i] ! myBBox
          PAR i=0 FOR SIZE in
            in[i] ? BBuf[i]
        -- Calculate max left, right, top, bottom section
        SEQ i=0 FOR SIZE in
          SEQ
            IF
              myBBox[Left] > BBuf[i][Left]
                myBBox[Left] := BBuf[i][Left]
              TRUE
                SKIP
            IF
              myBBox[Right] > BBuf[i][Right]
                myBBox[Right] := BBuf[i][Right]
              TRUE
                SKIP
            IF
              myBBox[Top] > BBuf[i][Top]
                myBBox[Top] := BBuf[i][Top]
              TRUE
                SKIP
            IF
              myBBox[Bottom] > BBuf[i][Bottom]
                myBBox[Bottom] := BBuf[i][Bottom]
              TRUE
                SKIP
    -- Printing section
    toMux ! 'L'
    toMux ! ':'
    out.number(myBBox[Left], 0, toMux)
    toMux ! ' '
    toMux ! 'R'
    toMux ! ':'
    out.number(myBBox[Right], 0, toMux)
    toMux ! ' '
    toMux ! 'T'
    toMux ! ':'
    out.number(myBBox[Top], 0, toMux)
    toMux ! ' '
    toMux ! 'B'
    toMux ! ':'
    out.number(myBBox[Bottom], 0, toMux)
    toMux ! '*n'
    SKIP
:
```

1.4 Result

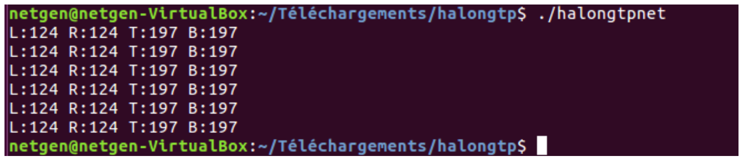

Figure 2: Display bounding box coordinator on terminal.

2 Find the maximum elevation

2.1 *How can we find the elevation of each cell ?

- **Step 1**: We need to create the behaviour file whose name is "nodes-test-include-cell.occ". In this file, we need to add two classes one is "CellNode" and the other is "Mux". "CellNode" represents each cell in the map, "Mux" is the method that displays all the output from "CellNode".

- **Step 2**: We need to get the elevation of each sensor by using the "CellNode" class in "nodes-test-include-cell.occ" file. To do that we can access to the global array "NetLocation" in "halongtp0.occ" and "halongtpData0.occ" files. In the "halongtpData0.occ" we can get all the parameter of the map through these parameters we can access to each "CellNode" in the "halongtp0.occ" so that we can get the elevation of each cell.

- **Step 3**: After we find the elevation, we need to use cooperation techniques to send the elevation to all sensor and find the max.

2.2 Code

```
PROC CellNode([]CHAN OF diam.proto in,out,
VAL INT Identity, CHAN OF BYTE toMux)
  -- Get all cells in the array
  CellArray myCell:
  -- Get position of each cell
  CellPosition myPosition:
  -- Get currently elevation
  REAL64 myElevation:
  -- Buffer is used to store all the elevations
  [MaxFanOut] REAL64 elvBuf:
  SEQ
    myCell := Cells[Identity]
    myPosition := myCell[position]
    myElevation := myPosition[elevation]
    SEQ Turns=0 FOR MaxNodes-1
      SEQ
        PAR
          PAR i=0 FOR SIZE in
            in[i] ? elvBuf[i]
          PAR i=0 FOR SIZE out
            out[i] ! myElevation
        SEQ i=0 FOR SIZE in
          IF
            myElevation < elvBuf[i]
              myElevation := elvBuf[i]
            TRUE
              SKIP
    out.real64(myElevation, 0, 0, toMux)
    toMux ! '*n'
:
```

2.3 Result

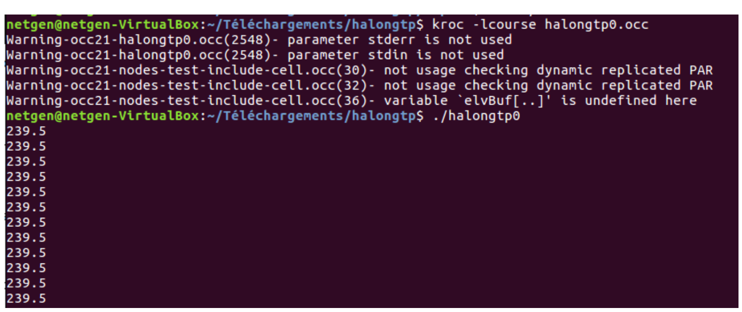

Figure 3: Display maximum elevation on terminal.