KFUPM

Senior Project design

COE485

Term 132

Tracking Employee Project

By:

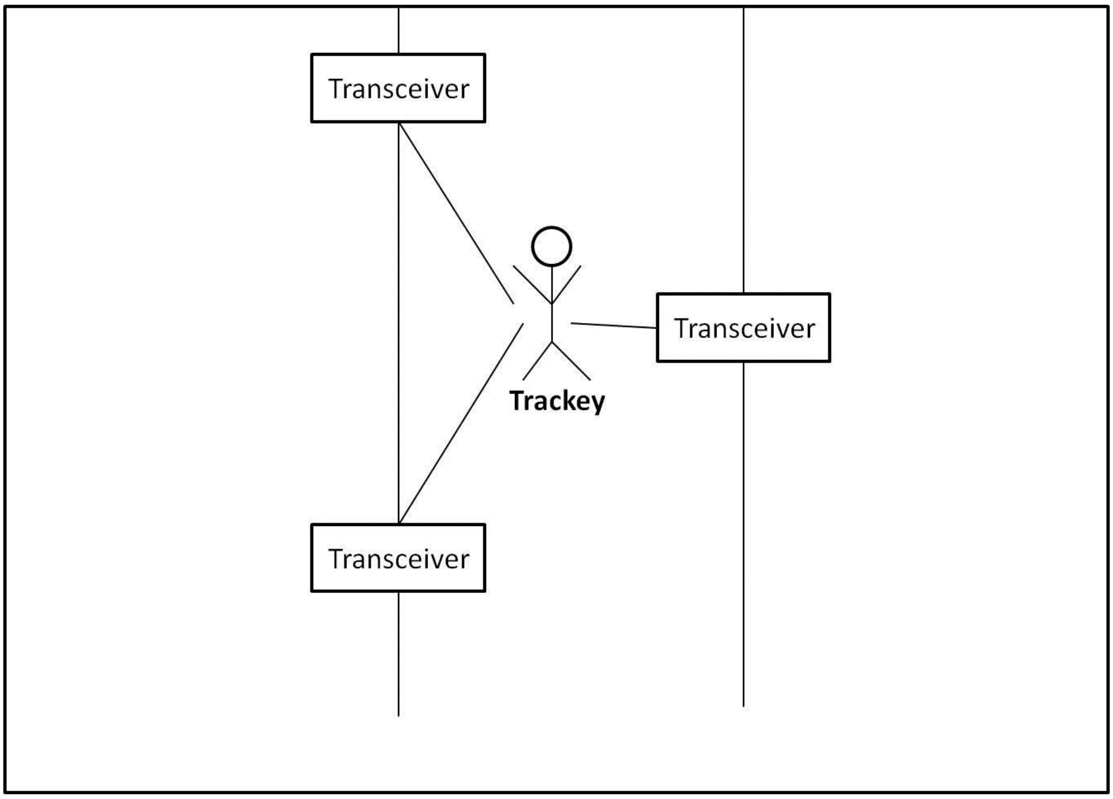
Abdullah ALGhaith

Mohammad AL-Hajji

Table of Content:

1. Introduction:

This project is designed to track employees or people within certain area coverage. The idea is to have a number of reference transceivers spread across a building, while the employees will be carrying a small device (Trackee) that is communicating with the transceivers and sending those signals constantly in order to be localized (see figure 1).



(Figure 1)

This can be used to localize the employees in emergency cases like fires, sinking, earthquakes and so on it is also can be used to monitor restricted areas.

2. Problem Statements:

The problem is that the companies sometimes needs a monitoring over the restricted areas and so on. In addition to the need of tracking the employees if a disaster happened like fires and so on. This proposed solution can be viewed from two different views; one way is that it may increase the efficiency of the employees and workers knowing that they are being tracked. Or it may be viewed as spying which might affect the moral of employees.

Why do we need this solution?

An offshore maintenance platform belonging to Saudi Aramco sank on December 27th, leaving three Saudi Aramco workers dead, according to a statement by Saudi Aramco.

"On Saturday morning, December 28, 2013, Saudi Aramco emergency response teams recovered the bodies of three contractor who drowned when the barge they were working on, the Arabia-4, sank in the Safaniya Offshore,"

Saudi Aramco failure to know where their employees are caused deaths and injuries to employees, If Aramco had a tracking system and the necessary precautions, this could have been prevented.

3. Project Specification:

The project should be able to show the position and the direction of the employees in real time.

The most two important specifications are the speed and accuracy. Speed because the system is tracking the employees in real time, specifically in emergency cases. Accuracy is need because we want to pin point the location of the employees within nearest meter.

4. Teamwork

Mohammad AL-Hajji

Responsibilities:

Serial Communication. Mohammad was the responsible for the communication between the gateway and the transceivers.

Contributions:

The algorithm that used to convert the RSSI to distance and the data base.

Expertise:

Microcontrollers and serial communication since he spent the Co-op period working on these things.

Abdullah ALGhaith

Responsibilities:

GUI (graphical user interface).

Contributions:

The algorithm that used to convert the RSSI to distance and the data base.

Expertise:

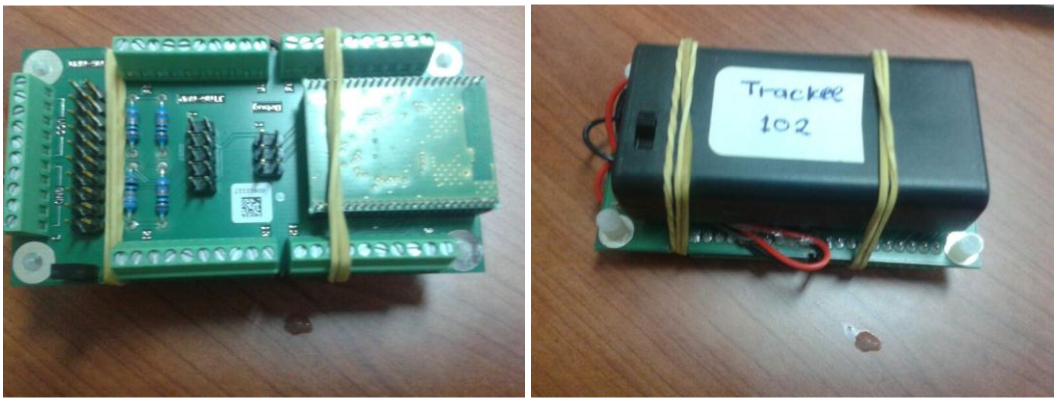
Java programming language.

5. Engineering design

5.1 Architecture

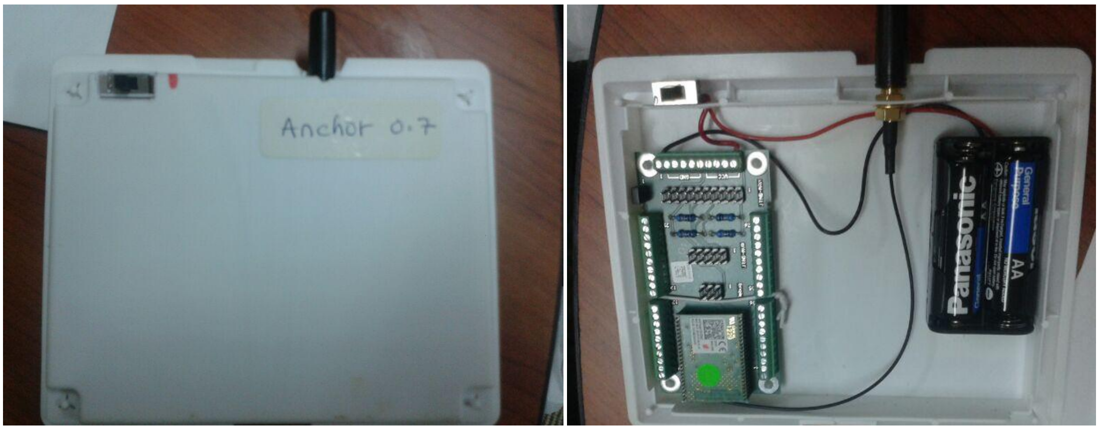
The system contains three things that communicate with each others to specify the position.

Trackee: the device is hold by the employee to know his position (see figure 2).



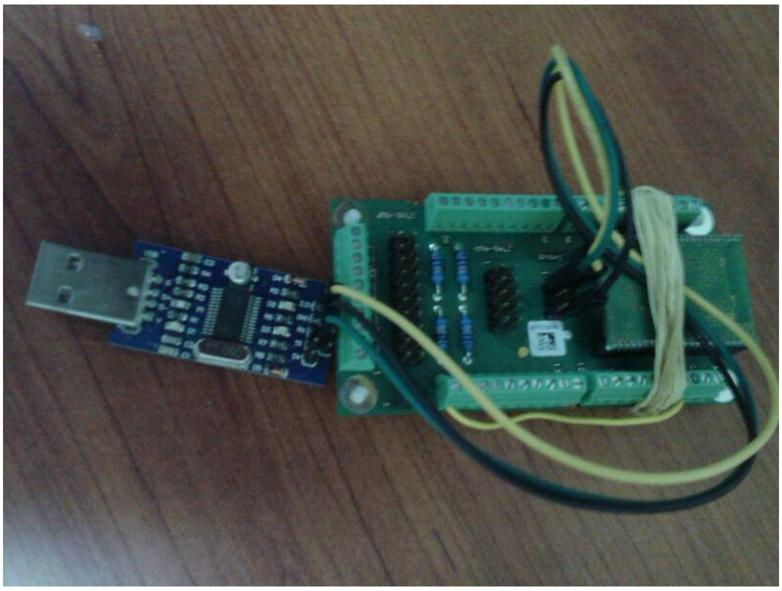
(Figure 2)

Reference Transceivers: the devices are fixed on the wall and communicate with the trackee to get its position (see figure 3).



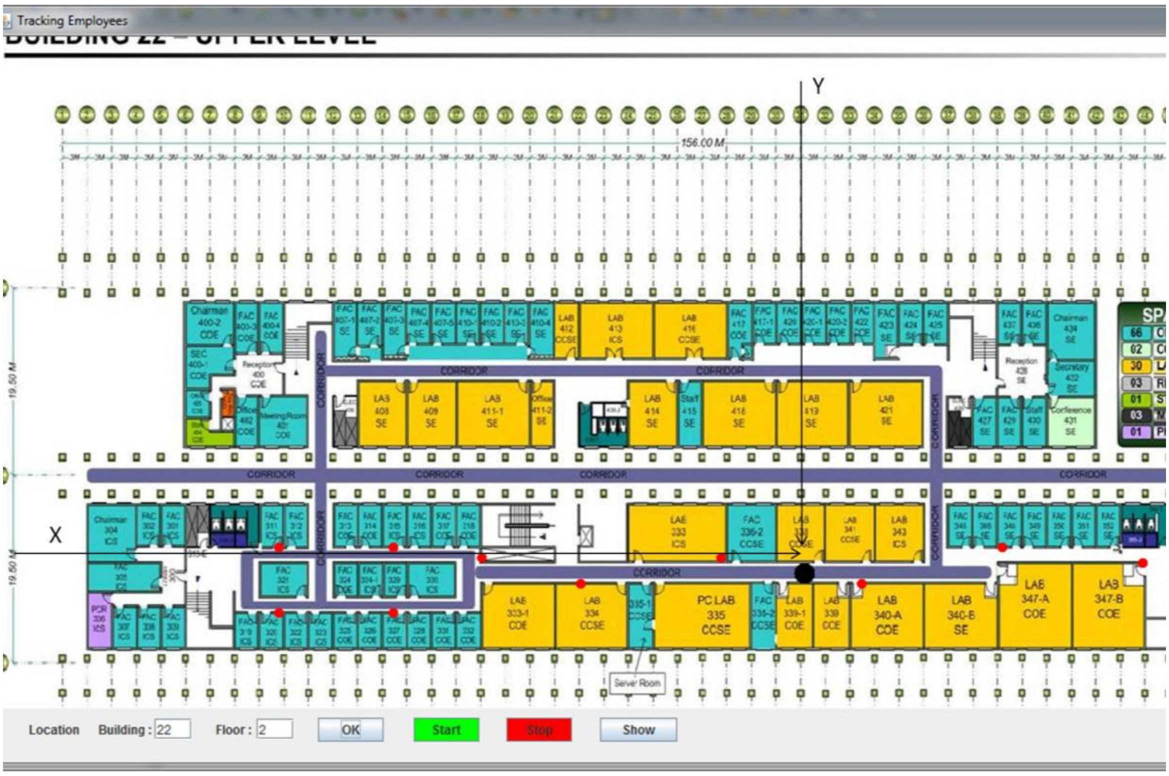
(Figure 3)

Gateway: the device is connected to the computer (gateway) and communicate with the transceivers to get the RSSI values that are used for position determination (see figure 4).



(Figure 4)

Software part is the programming part. Eclipse used for designing the GUI in java (see figure 5).



Building : 22

Floor : 2

(Figure 5)

There are two text fields used to insert the building number and the floor for example building 22 floor 2 as shown in the figure 5, after pressing OK button the map will be shown along with the transceivers (the red points on the map). The green button Start is used to start the application by changing the x, y value according to the received value from the transceivers and show it as a black circle. Show button is used to show the lines from the trackee to the transceivers that communicate with it. Finally, the stop button is used to stop the tracking.

The process start from the trackee, it sends continues signals to the transceivers within its range. The transceivers get the signal and put the value in a packet

(#, Trans0, Trans1, Tr0, Tr1, Serial#, RSSI, LQI, #)

Where:

trans0 & trans1 are the transceivers that send the packet to the gateway named (01, 02,...).

Tr0, Tr1 are the trackee that send the signal to the transceivers named (01, 02,...).

Serial# is the serial number of the packets start from 1 to 255 then start again from1.

RSSI is the received signal strength and it is the number that used to calculate the distance between the transceiver and the trackee.

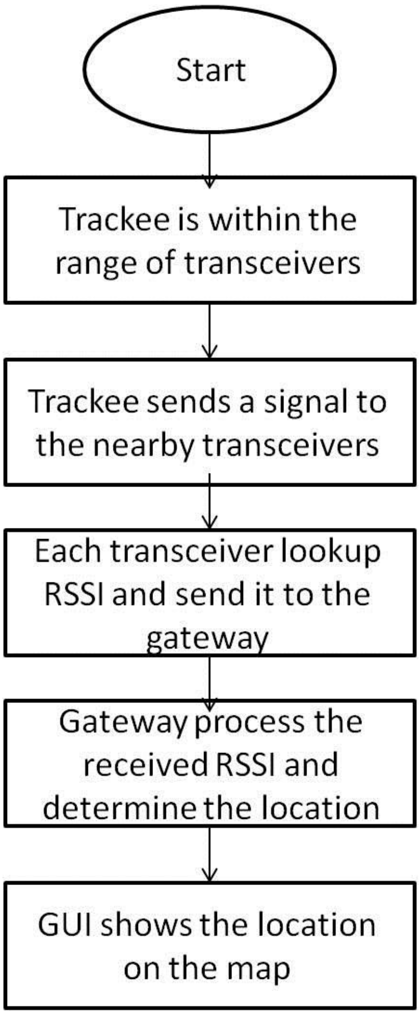
LQI is the link quality indication its value is 255 and it may change according to noise nearby.

#, # the hashes are used to make sure that the packet received is complete.

5.2 Design decision

There are different designs that can be used for such a project. First design is to use 2 transceivers for determining the position but the accuracy will not be as good as needed. Second design is to use three transceivers and use the Trilateration algorithm and that's the one we used since it is much accurate than the first design. Third design is to use 4 transceivers but there weren't a big different from the use of three transceivers in addition to that it needs more transceivers to work.

5.3 Component Design and Implementation


Start
Trackee is within the range of transceivers
Trackee sends a signal to the nearby transceivers
Each transceiver lookup RSSI and send it to the gateway
Gateway process the received RSSI and determine the location
GUI shows the location on the map

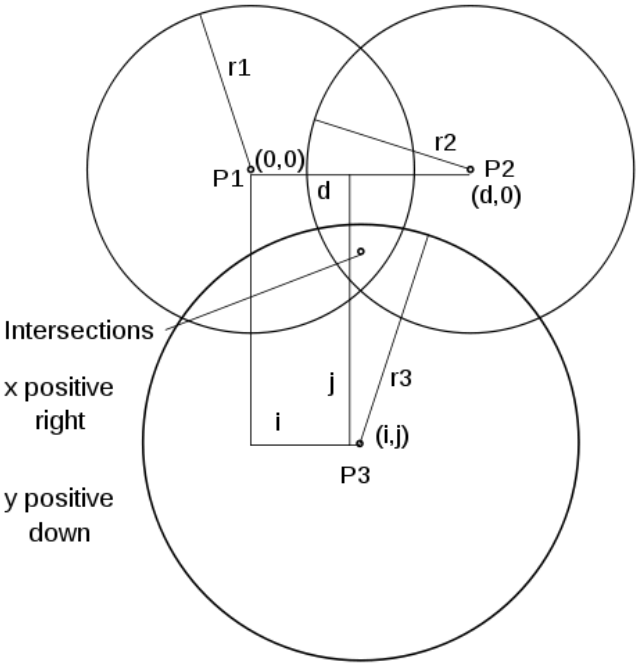
(Figure 6)

Figure 6 shows a flowchart of the system. All the components used such as the atmega and the communication protocol are off the shelf (ready to be used).

5.4 System Integration:

For demo purpose, we used three transceivers and one trackee within the lab area and try to apply the algorithm. The testing was as follow:

1. Put the three transceivers on their fixed location as shown below (P1, P2, P3).

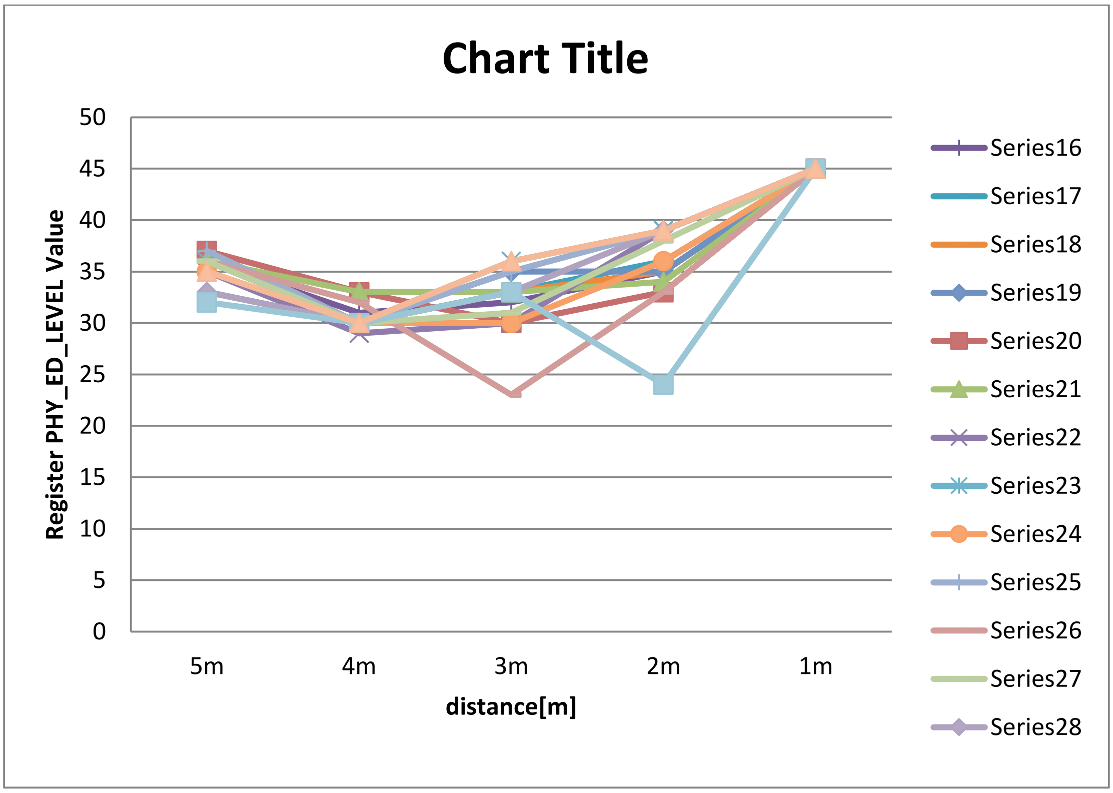


(Figure 7)

2. The gateway that connecting directly to the computer will start receiving packets and extract the information and send it to the algorithm in addition to send a copy to the database.
3. The algorithm will convert three RSSI's received from the nodes into radius r in meter due to the relation on (Figure 8) (Table 1) and do some calculations to localize trackee's position (x, y) within their ranges.
4. The result will be sent to GUI interface to display the trackee coordinates.

	1m	2m	3m	4m	5m
read1	45	35	32	31	36
read2	45	36	33	30	36
read3	45	35	33	30	37
read4	45	35	35	30	33
read5	45	33	30	33	37
read6	45	34	33	33	36
read7	45	39	30	29	35
read8	45	39	36	30	36
read9	45	36	30	30	35
read10	45	39	35	30	37
read11	45	33	23	32	36
read12	45	38	31	30	36
read13	45	39	33	30	33
read14	45	24	33	30	32
read15	45	39	36	30	35
AVG	45	35.6	32.2	30.53333	35.33333
MIN	45	24	23	29	32
MAX	45	39	36	33	37

(Table 1)



(Figure 8)

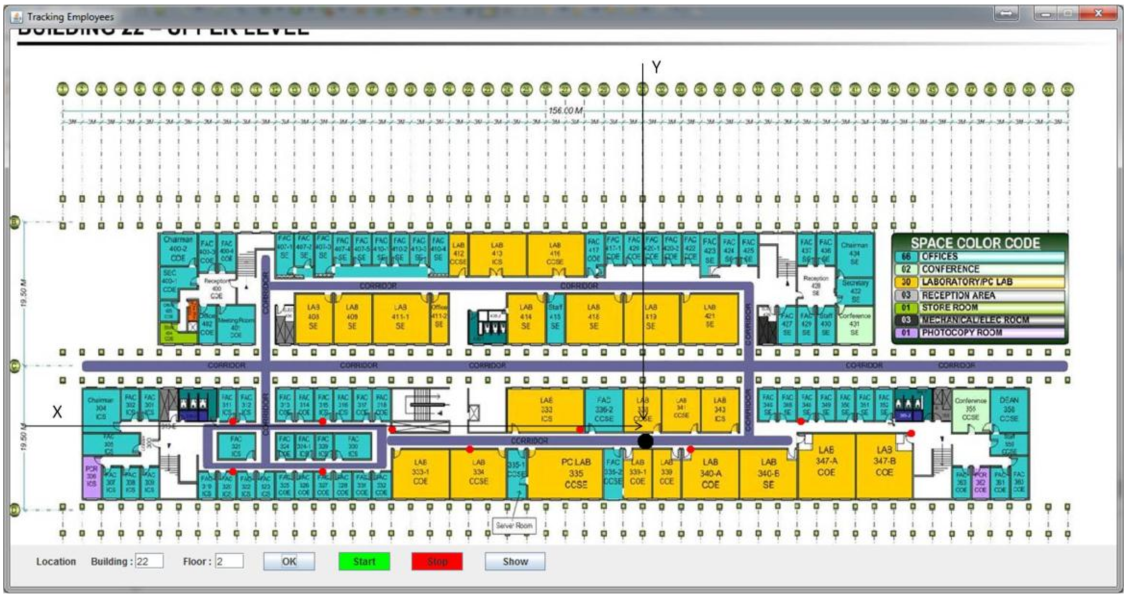
6. Issues:-

First problem is the initial RSSI that the gateway received from the transceivers. We tried to find a relation between the RSSI received and the distance between the trackee and the transceiver but we got a weird (x, y) position. After that we figured out that we have to do some processing on the RSSI before using it but the problem was still there. Finally, we found a solution, which we have to get a number of values and take the average for them and then use the average in the algorithm.

Second problem was in the GUI; we couldn't draw the transceivers and the trackees at the beginning. The problem was because we plug the map as a component. The solution was just by changing the map into background and draw the trackees and transceivers as a component on top of it.

Checking the battery of the trackee is a most between time to time to avoid getting bad results because of the low power and that is one of the limitation. Another problem is the short range for the transceiver which means for a big buildings a large number of transceivers have to be used.

The design limitation is that the display is in 2D only. Since is the first time this system is implemented, the focus wasn’t on visualization, but it was on implementing the algorithm and getting the most accurate position of the trackee (see figure 8).



Location Building : 22 Floor : 2

(Figure 8)

7. Engineering Tools and Standards

The microcontroller used is Atmel ATMEGA128rfa1, base on IEEE 802.15.4. There are many types of microcontrollers that can be used such as ARDUINO, TELOSB, MicaZ and so on but the main reason behind our choice is that the ATMEGA contains the MCU and the Radio embedded in a small chip, while for the ARDUINO extra shield need to be installed on top of it for communication. Another reason of using the ATMEGA is that it is the one widely used in the industries.

For the communication protocol, we used IEEE802.15.4. There are of course many choices like RFID and ZigBee, however, RFID has a short range and many transceivers have to be used, which is costly. Moreover, RFID can't determine the exact position. For the ZigBee, it is not an open source so it is not free. In addition, it consumes more power than Atmel ATMEGA128rfa1. Moreover, ZigBee is using AODV in the network layer, which is hard coded. However, in Atmel ATMEGA128rfa1, the routing protocol in the network layer is an open source and therefore this will save energy.

The programming language that was used for the GUI is JAVA because it satisfy the needed in such a project such as being user friendly

8. Conclusion

This project was about designing a system that is able to track and localize employees or people within a certain area, which might as well be helpful in emergency cases such as fire, sinking, earthquakes and so on.

We have learned how to perform serial communication between nodes, how to program these nodes to send packets, and how to correlate RSSI with distance. In addition, we learned how to build a database for industrial needs.

9. Appendices

The code for loading maps and taking care of error given values for the maps is:

```
if(actionCommand.equals("OK")){
                try{
                    floor=Integer.parseInt(TFloor.getText());
            building=Integer.parseInt(TBuilding.getText());
                    if(building==22&&floor==1){
                            //MAP 1
                        map1=true;
                        map2=false;
                        repaint();
                        JOptionPane.showMessageDialog(null, "The
map 22-1 loaded successfully");
                    }
                    else if(building==22&&floor==2){
                        //MAP 2
                        map2=true;
                        map1=false;
                        repaint();
                        JOptionPane.showMessageDialog(null, "The
map 22-2 loaded successfully");
                    }
                    else{
                        JOptionPane.showMessageDialog(null, "The
map you entered is undefined !");

                   }
                 }
                catch(Exception g){
                    JOptionPane.showMessageDialog(null, "please enter an
integer number representing the location");

                    }
                }

public void paint(Graphics g){

            super.paint(g);
            if(map1){
            Image
a=Toolkit.getDefaultToolkit().getImage("C:\\Users\\maaf4ever\\Desktop\\1.jpg");
            g.drawImage(a,0,0,getSize().width,670,this);
            g.setColor(Color.RED);
        }
             if(map2){
            Image
a=Toolkit.getDefaultToolkit().getImage("C:\\Users\\maaf4ever\\Desktop\\2.jpg");
            g.drawImage(a,0,0,getSize().width,670,this);
            g.setColor(Color.RED);

            g.fillOval(1100,525, 10,10);
            g.fillOval(965,510, 10,10);

            g.fillOval(830,545, 10,10);
            g.fillOval(695,520, 10,10);
            g.fillOval(560,545, 10,10);
            g.fillOval(465,520, 10,10);
```

```
                            g.fillOval(380,573, 10,10);
                            g.fillOval(380,510, 10,10);

                            g.fillOval(270,573, 10,10);
                            g.fillOval(270,510, 10,10);

                    }
```

The code for the GUI design is :

```
public TrackingEmployee() {
                    super("Tracking Employees");
                    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
                    setSize(frame_w, frame_h);
                    setResizable(false);
                      getContentPane().setLayout(null);
       JLabel LLocation = new JLabel("Location");
       LLocation.setBounds(32, 658, 57, 14);
       getContentPane().add(LLocation);

       JLabel LBuilding = new JLabel("Building :");
       LBuilding.setBounds(99, 658, 63, 14);
       getContentPane().add(LBuilding);

       TBuilding = new JTextField();
       TBuilding.setBounds(153, 655, 36, 20);
       getContentPane().add(TBuilding);
       TBuilding.setColumns(10);

       JLabel LFloor = new JLabel("Floor :");
       LFloor.setBounds(212, 658, 47, 14);
       getContentPane().add(LFloor);

       TFloor = new JTextField();
       TFloor.setColumns(10);
       TFloor.setBounds(251, 655, 36, 20);
       getContentPane().add(TFloor);


       JButton BStart = new JButton("Start");
       BStart.setBackground(Color.GREEN);
       BStart.setBounds(402, 654, 63, 23);
       getContentPane().add(BStart);

       JButton BStop = new JButton("Stop");
       BStop.setBackground(Color.RED);
       BStop.setBounds(491, 654, 63, 23);
       getContentPane().add(BStop);

       JButton BOK = new JButton("OK");
       BOK.setBounds(310, 654, 63, 23);
       getContentPane().add(BOK);

       JButton BShow = new JButton("Show");
       BShow.setBounds(581, 654, 74, 23);
       getContentPane().add(BShow);
       BStart.addActionListener(this);
```

```
        BStop.addActionListener(this);
        BShow.addActionListener(this);
        BOK.addActionListener(this);
        setVisible(true);
```