

University of Pennsylvania
Electrical and System Engineering

**ESE 519 Real Time Embedded Systems
2013 Fall**

Instructor: Prof. Rahul Mangharam

Final Project

REAL WORLD “Tank CITY”

Final Report for Group:

Wei-Ting Lo

Peng Hao

Yun Yuan

Date: Dec 20, 2013

Blog link: <http://tankbattlecity.blogspot.com/>

Youtube link: <http://www.youtube.com/watch?v=ERtUoUajkBM>

Github link: <https://github.com/ESE519/TankCity>

Part 1 Motivation and Goals

Most of us have played a game called “Tank city” on PC when we were little kids. Our team is going to build the whole scenes in the reality world.

For this real game, we will have enemy tanks, user tank, base flag, and a real map.

There is one player that is controlled by a user. It can moves around and shoot the enemy tanks. We use white LEDs light to simulate the “shoot” and use light sensors equipped on the enemy tanks to detect the when the tank is being shot. If enemy tanks get shot, they will lose HP value. If all HP Value of them are gone, they dead. If all enemy tanks are dead, player win the game.

There are going to be two enemy tanks, which would run automatically. Their goal is to go to the base and attack it. If they arrive the base and hit it, the user lost the game. It is like a tower defense game. User need to try their best to stop the tanks.

We design a base and some lanes for the map. We decorated and made walls, which are made of cardboards, as barrier.

This is an integrated project. We utilized many techniques here, including microcontroller, sensor system, analog and digital circuit, and network technology.

Part 2 System Design

The whole system mainly consists of three parts: Player Tank Control System, Enemy Tanks Automatic System, and Location and Map Navigation System. All three parts have their own necessary features and collaborate to get the whole system work.

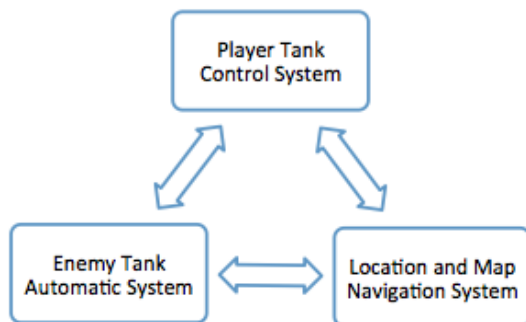


Figure 1. The whole system

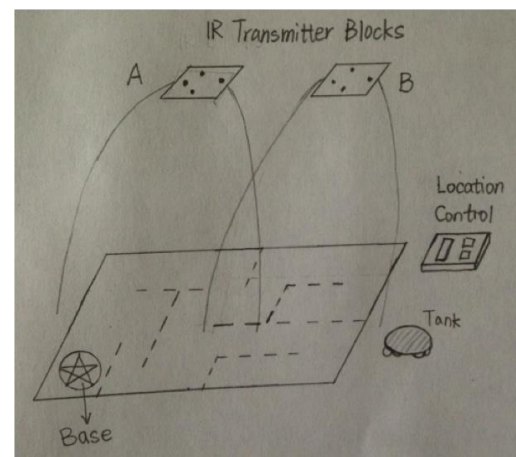


Figure 2. The network for the location

There is a network between Enemy Tanks Automatic System, and Location System.

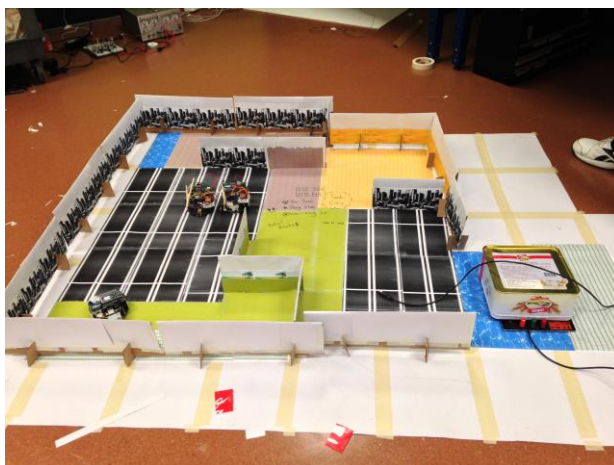


Figure 3. The whole map for the project

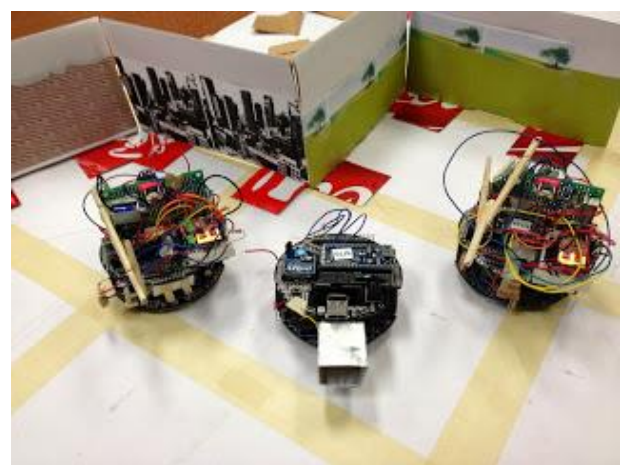


Figure 4. The tank family

The location system has 2 IR transmitter blocks, let's just call them A and B, on the ceiling. Since there is a distance limitation for the IR location system, we set up two blocks in order to cover the whole map. The A system covers the left part of the map, and B covers the right side of the map.

If the enemy tank is in area A, it uses block A to navigate the tank. When the tank reaches the limit of area A, the location system switches to B and shut down A.

The Hardware we used in this project:

m3pi robot (3);
mbed (5);
joystick (1);
Xbee (2);
White LEDs;
ZigBee (4);
Light sensors;
Resistors with different values;
Relay (4);
Batteries;
Solenoid (2);
IR camera (2);
5 V power supply;
IR LEDs;
Speaker;
firefly sensor board;
green LEDs;
wires.

The software IDE we use in this project:

Keil uvision4
Putty as terminal and input for the XBEE.

Part 3 Hardware and Software Detail

1. User Tank Control System

The tanks we are using are m3pi robots from pololu.

We designed a remote controller, which has a joystick and a button, for user to control the user tank. Joystick as moving the tank and button as shooting.

The idea of controlling the tank is:

Control joystick or button->send analog data to mbed->mbed determine the data and output digital signals to XBee->XBee tells the XBee on the tank which digital signals it has received from mbed->XBee on the tank tell the mbed on the tank to move according to the digital signal it has received.

By moving the joystick leftwards, rightwards, forwards and backwards, it will output different voltage to a microcontroller, mbed from NXP.

The mbed processes this input and send data to the user tank via XBee.

The user tank, has a mbed and XBee on it. We use mbed to control every decision on the tank, and use XBee as communication.

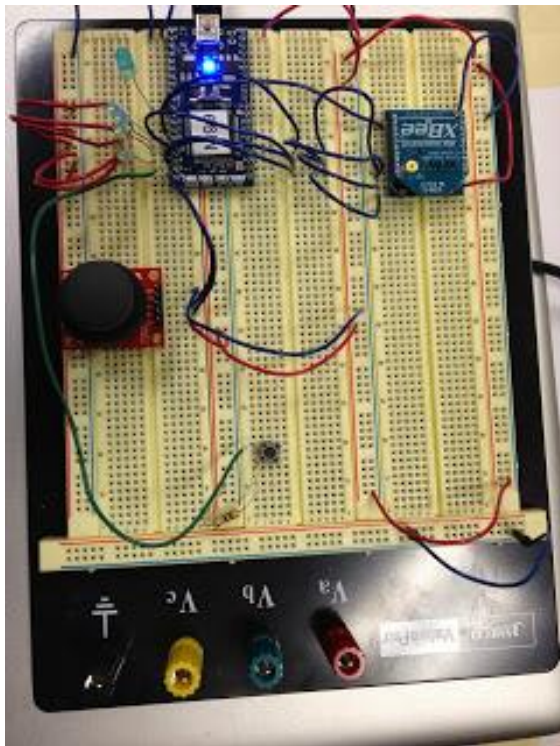


Figure 5. Joy stick control panel

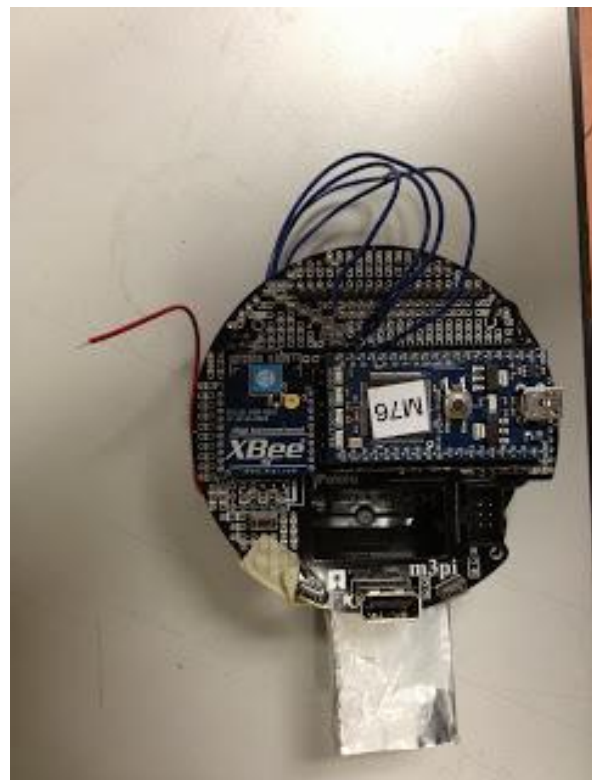


Figure 6. User tank with "cannon"

2. Location System

A. Hardware summary

The central control system consists of one mbed, one MRF24J40 (ie ZigBee RF Transceiver Module (2.4 GHz), to communicate with the whole system), two relays, and one speaker. This system is mainly responsible for GPS localization tiles switch and communication with the enemy tanks.

For the tiles switch part, we have 2 IR transmitter blocks on the ceiling with 4 IR LED spots each with 5V power supply. We use two relays to control power supply for different IR transceivers and controlled by mbed. And the MRF24J40 is used by both the central control system and enemy tanks to communicate to know when the tiles need to switch. When the central control system is told to switch a tile, it sets the respective pin to high and so that the corresponding relay will supply for that tile.

The central control system also has another functions that it will alert the speaker when the enemy tanks attacks the tank base. This is implemented by the communication between the tank base board and the central control system.



Figure 7. IR blocks on the ceiling

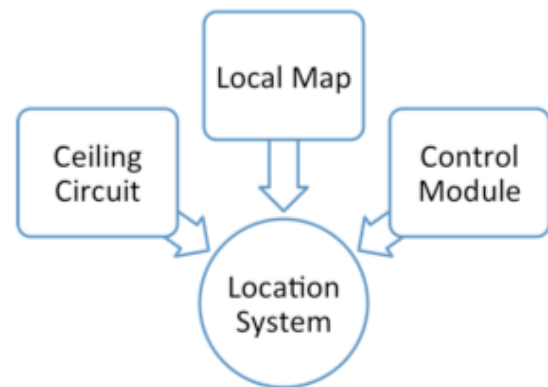


Figure 8. Design block for location system

As mentioned before, the enemy tank would request IR block switch when it arrives at the boarder of area under current IR coverage.

Then the enemy tank would send a request to the location control system and wait. After receiving the request, the control system switches the working IR Block to another with relays, and sends an ACK to the request enemy tank. After receiving the ACK, the enemy

tank knew that the IR Blocks have switched and continued to move under the new coordinate.

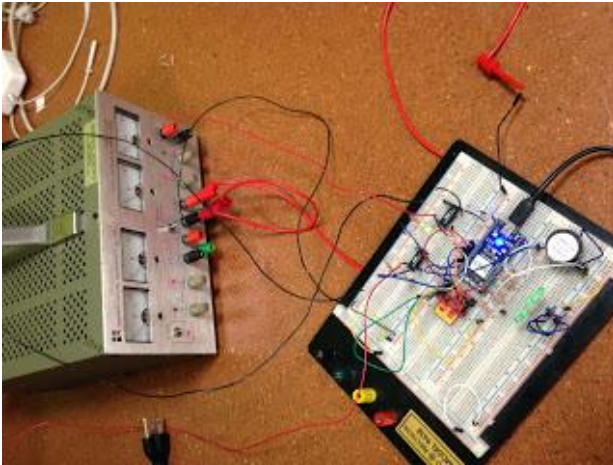


Figure 9. The switch system

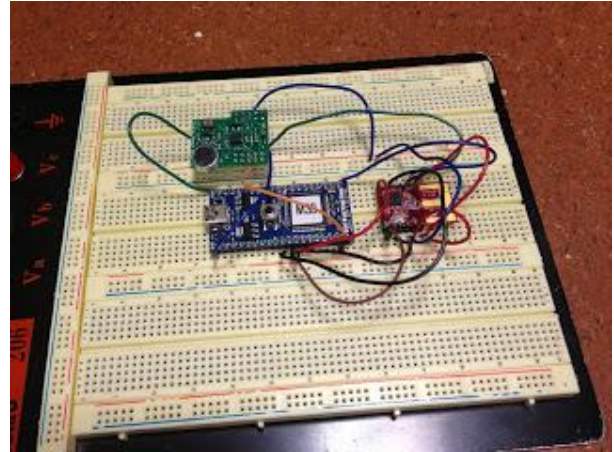


Figure 10. The base circuit

The tank base system consists of one mbed, one MRF24J40, and one firefly sensor board. For the firefly sensor board, we used the sound sensor to determine whether the base is being attacked or not. Since our enemy tank will shoot a rubber bank and when it hits the base, it would have sound for the sound sensor. After the sensor detects a sound signal, it would send a signal via ZigBee to the central control system to alert the speaker. Once he speaker sounds, the user lose the game.

B. Software summary

In the coding part, we have redesigned the transmission and receiving function according to the MRF24J40 library. Two functions named `rf_send` and `rf_receive` are used to send and receive signals.

On the Central Control System:

For the enemy tank, when it arrives at the boundary of two tiles, it will call a function, `calibrate`, to switch the location system. It includes the command in the transmitting package and send it to central control system.

The central control system will listen signals all the time, when it receives a signal, it will parse the command. When it receives the “switch” command, it will set the corresponding pin high to switch different tiles. And when it receives a “Finish” signal that means the game is over and it alerts the speaker.

On Base:

Determinate whether the base is being attacked or not. If yes, send a signal to central control system and ask it to announce the game is over. If no, stays the same.

3. Enemy Tank Automatic System

A. Hardware summary

We have built a sophisticate Enemy tank automatic system.

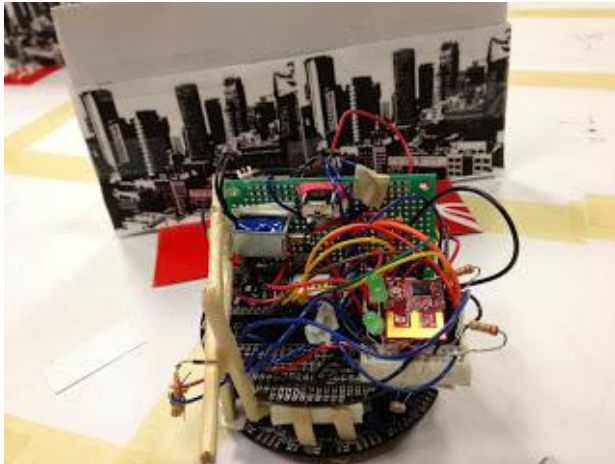


Figure 11. The enemy tank with full function

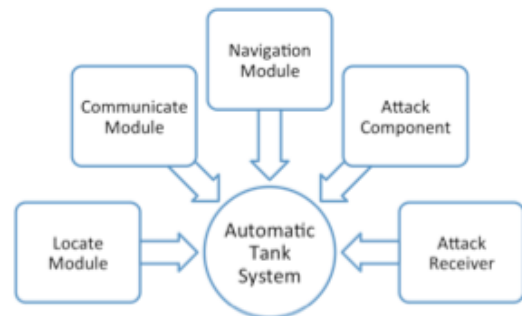


Figure 12. Design blocks for automatic enemy tank

The enemy tank use a mbed as its main controller.

How does the tank moves:

A Wii IR camera equipped on the enemy tanks is used to analysis the relative x and y coordinate according to the IR location system.

Since we have four group of IRs in a set, the tank will receive four different xy coordinate. These 4 groups of data can be used to calculate the absolute location of the tank on the map. The enemy tank use this information to navigate and guide the tank. We use two of the IR groups as angle reference. Let's say we pick group 3 and 4. The calculation to find the reference angle is: $\text{atan2}((x_3-x_4), (y_3-y_4))$. Once we have xy coordinate and angle, moving the tank to a desired destination is convenient.

How does the enemy tanks know they are being attacked:

There are light sensors equipped on the sides of enemy tanks, so they can detect "shoots" from the user tank. Once the light sensors detect the lights, it would reduce the health point of the tank.

How does it attacks:

There is a physical shooting unit on the enemy tank to attack the base. We designed a simple rubber band gun and install in on the tank. If the enemy tank arrives the destination, in front of the base, it will start shooting the rubber bank by using solenoid as trigger.

B. Software summary

On Enemy tanks:

We used nested if statement to program the tanks.

Basically, each if statement is like a check point for the tanks. Once the robots reach the destination, it will close the current loop and goes to the next one. At the mean time, the tanks would also constantly check their current angles to make sure they are moving straight. For the first tank, when it reaches the center of the map and switch tile, we uses new tile systems as reference to move. Hence, It needs to use totally different angle and xy coordinate as reference to move.

For different tanks, their status change as following.

Tank 1: Starts first -> move to the center of the map -> use ZigBee to call for switching tile -> move to the base->attack the base.

Tank 2: Starts second (after the tile switch) -> move to the base -> Attack the base.

Part 4 Discussion and Conclusion

After we decided to do the project, we used top-down design idea and used it to direct our whole development process.

Separate the whole system into 3 different modules at the first and integrate them together in the end, which makes our design more reasonable.

For each modules, we first build a test system on breadboards. On the breadboard, we adjust parameters and circuits. Once we were sure everything was correct and good to go, we transfer the test system to the real device. After done transfer, we test each of them and see if they were working fine as they did on the test system. In the end, we integrated each of the modules together, and tested all of them. We found this method is good for our project. By using top-down idea, we keep checking everything before implement them in the real devices. In addition, it is a good way to debug as well.