Team Name: NewBUPTERS
Project Name: YUN-IRF
Team Members: Xu Zhang, Yijing Zeng, Haichuan Guo, Haofu Liao
Team Leader: Haofu Liao
University: Beijing University of Posts and Telecommunications
Address: Xi Tu Cheng Road, Haidian District, Beijing, No.10
Contact Tel Number: 15001118363, 13426013833
Postal Code: 100876
Email: liaohaofu@gmail.com, wjzx1120@bupt.edu.cn

The way leading to New York
Imagine a world where technology helps solve the toughest problems facing us today
1. Executive Summary

The World Health Organization estimates that in 2011 one-third of the world is under-fed and one-third is starving. It is reported that some 800 million people in the world suffer from famine. Every day, almost 16,000 children die from hunger-related causes. That’s to say, one child will die every five seconds. The word famine means the complete absence of food. Famine and hunger are not new problems, nor are they confined to the third world. People have been looking for the solutions to hunger for centuries but famine still exits. According to Action-Against-Hunger-Organization’s Analysis, the causes of absence of food are shown in fig1.

As fig1 shows, we can conclude that it is inefficient irrigation and water shortage that mainly cause the absence of food. Also the Guardian reports that if we continue to use inefficient irrigation methods, the continents might be able to feed just 35% of its population by 2030.

It is high time we applied an efficient irrigation system to farmland. However neither flood irrigation nor drip irrigation, which is the main irrigation systems today, can irrigate reasonably. These irrigation systems don’t consider the actual local weather and the actual type of crops. Farmers don’t give what the crop needs, and it is hard for farmers to control the irrigation system in real-time. Our YUN-IRF just deals with these issues by controlling watering and fertilizing automatically according to the actual local weather and type of crops.

For developed and less developed regions we designed two systems, which are the YUN-IRF and the IRF.

In less developed regions our irrigation system-IRF consists of two parts: the master system and the slave system. The slave system consists of several Zigbee routers (the number is determined by the scale of the farmland), temperature and humidity sensor, relay circuit and solenoid valves (a kind of switch which controls liquid flow). The routers distribute in different parts in the farmland and can collect the temperature and humidity, transmit them to the master, receive commands from the master respectively and control the solenoid valves accordingly.

IRF’s master system includes an eBox, a LCD screen and a Zigbee coordinator. Users initialize the system under the guide on the LCD screen, including initializing the kind of crop, the farmland’s location in the world and the growth stage of the crop. eBox is responsible for running automatic controlling algorithms according to the initialization information and the real-time data collected by Zigbee routers. Zigbee coordinator transmits the data received from Zigbee routers to eBox and transfers commands from eBox to Zigbee routers.

In developed regions IRF can be upgraded to YUN-IRF. We transfer most of the storage and calculation tasks to Windows Azure, which is also regarded as the concept of Cloud Computing. Taking advantages of the powerful computing and the mass storage capacity of the Cloud Platform, we store the historical data of the farmland. On the basis of these historical data and more practical factors, we put into use more scientific algorithms which are Penman-Monteith equation & Mitscherlich-Bray model. User interface is deployed in the Windows Azure as a B/S application and user can gain access to it by various equipments (such as PCs, cell phones, PDAs etc), meaning users can control YUN-IRF whenever and wherever possible.

2. Problem Analysis

According to the International Food Policy Research Institute’s report in 2010, despite some progress in fighting world hunger, there’re still 29 countries have alarmingly high levels of hunger. As our state in the summary part, improving the efficiency of irrigation system is the best method to get rid of famine. However it’s hard for one system to apply to
different types of climate. What’s more, crops are various from different region in the world, thus makes it difficult for one system to apply to various crops. As shown in the fig below (statistical data from IFPRI) famine almost throughout all continents including most types of climate. These issues make it hard to develop an effective irrigation system which can be applied to regions all over the world.

3. Project Analysis

3.1 Solutions to the Problem

Solving famine might seem to be a challenge, but it can make a big difference that every farm in the world irrigates their crops using as little water as possible and fertilizes them much more scientifically, such as taking our intelligent irrigation systems-IRF. With YUN-IRF or lite edition-IRF farmers can use water as little as possible to get the maximum effectiveness of irrigation. If most farmers in the world apply intelligent irrigation system to their farm, then the global food production will increase and food price will decrease. Finally famine is expected to be relieved.

3.2 Innovation

3.2.1 Automatic control

Taking into consideration the fact that most farmers in the world have a relatively low knowledge level and usually don’t know much expertise about irrigation, we apply automatic control algorithms to our system and users only need to know the kind of crop, the location of the farmland in the world and the growth stage of the crop, and click the button to gain automatic control. Our system will begin to automatic control the proportion and the amount of fertilizer and water flow.

3.2.2 Intelligent Fertilization

Despite the crop’s demand for fertilizer is not the same in different growth stage, but the period of different growth stage is fixed in a year, and in one growth stage the crop’s absorption rate of N (P or K) is basically the same. Our equipment will judge the crop’s current growth stage according to the current date, and adjust the proportion and the amount of fertilizer automatically. We will pre-store different fertilization patterns of different crops in embedded devices and even Cloud Platform. Users only need to select the kind of crops on the screen.

3.2.3 Intelligent water flow control

Most farmers understand that the crops can grow well as long as there is water but a crop’s actual demand of water is mainly determined by temperature, humidity, the growth stage of the crop according to a research about the relationship between water supply and crop growth. Therefore we take temperature, humidity and the growth stage of the crop as decision variables, modeling the water flow and at last we conclude a scientific water flow control
algorithm. All the users’ control operations are only clicking the screen few times.

### 3.2.4 Zigbee Router Algorithm Make the System Easy to Deploy

Instead of using traditional cable communication method based on Rs232 between sensor and embedded device. We use Zigbee technology and designed a router algorithm to achieve it which greatly simplifies the installation process. As for different scale of planting area, we only need to correspondingly increase or decrease the number of sensors.

### 3.2.5 Use Cloud Servers to Make the Network More Stable

Instead of using one personal server, the Cloud uses hundreds, thousands or even more inter-connected server that are running at a higher speed constantly, affording a stable network and more bandwidth for our system, thus making the network safe to for all users.

### 3.2.6 Move Computing Tasks to the Cloud Servers

We move most of the calculation tasks to the Windows Azure. It has powerful computing capacity. We add more practical functions and implement more complicated automatic control algorithm on it, thus upgrading our system to YUN-IRF.

### 3.2.7 Make Use of the Cloud Server’s Sufficient Capacity

We establish the databases of global climate and the need of fertilizer for different crops on the SQL Azure. Users only need to select the region in the global map and the kind of crop.

### 4. User Experience

#### 4.1 Lite Edition—IRF.

**Easy to Deploy**

We give special attention to the convenience of installation. Basically, users only need to turn on nodes to complete the configuration, and other works, such as registration and configuration, will be completed by the system automatically.

**Friendly User Interface**

As an embedded device, everything is on the screen and the users only need to select the kind of crop, the growth stage of the crop and the farmland’s location in the world which are presented by pictures. Then the system will fertilize the crop and control the water supply automatically.

**Effective Control**

By apply intelligent Fertilization and water flow to the eBox, IRF can make full use of limited resources to get the maximum benefits.

#### 4.2 Upgraded Edition—YUN-IRF

On the basis of IRF, YUN-IRF gives users more excellent experiences.

**More Effective Control**

By taking into consideration of the historical record and adding the databases of global climate and the need of fertilizer for different crops, we can get a more scientific proportion and amount of N, P, K and water.

**More Convenient**

We afford an interface built on the Windows Azure. Users can visit it through web pages and are not required to install anything. So you can access it by various equipments (such as PCs, cell phones, PDAs etc), meaning that users can control YUN-IRF whenever and wherever it’s possible.

### 5. Market Analysis

#### 5.1 Reasons for Users to Use IRF

In the beginning of our project, we do a research on farmers about what is the main reason why farmers don’t apply intelligent irrigation system to their farm. The result is shown in the fig below.
Our system is just designed to solving these problems.

- By collecting previous temperature and humidity information from the different parts of the farmland, we apply intelligent irrigation algorithm to maximize the efficiency of irrigation.
- By implementing this system on an embedded device, we cut down the cost of the system to about 500 USD comparing to using PC hardware. Also the upgraded edition—YUN-IRF put the Windows Azure to act as server, thus cutting down the cost of the system comparing to using personal server. What’s more the cost can be cut down further when more people buy our system. At that moment, different users share the Cloud Server which lowers personal cost. Furthermore, we can make cooperations with some hardware vendors directly to lower the cost further.
- Using Zigbee technology to achieve communication between sensor and terminal device instead of using cable communication method. When deploying the system the first time or expanding the scope of the farmland, users only need to put a Zigbee router in the correspondingly region and turn on the switch and then it will be registered on the system automatically.
- Instead of using one personal server, Cloud Server affords a stable network and more bandwidth for our system.
- By building the user interface on the Windows Azure, users can control YUN-IRF whenever and wherever he has a cellphone.

5.2 Market position of IRF

In order to know the potential market position of our systems in IRF family, we compare our systems with other irrigation systems. The result is shown in the fig below.

<table>
<thead>
<tr>
<th></th>
<th>Cost</th>
<th>Efficiency</th>
<th>Convenience</th>
<th>Stability</th>
</tr>
</thead>
<tbody>
<tr>
<td>YUN-IRF</td>
<td>Mid</td>
<td>High</td>
<td>High</td>
<td>High</td>
</tr>
<tr>
<td>IRF</td>
<td>Low</td>
<td>High</td>
<td>Mid</td>
<td>Mid</td>
</tr>
<tr>
<td>Ari-See</td>
<td>High</td>
<td>Mid</td>
<td>High</td>
<td>Mid</td>
</tr>
<tr>
<td>Intertec.io</td>
<td>Mid</td>
<td>Mid</td>
<td>Low</td>
<td>Mid</td>
</tr>
<tr>
<td>Lingqi</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>

After comparison we can conclude that for the low-end market, our IRF can have a position in this market with its much lower price and relatively higher performance. At the same time for the high-end market, our YUN-IRF can be superior to any other similar products with excellent performance of all aspects, which can take a place in this market.

6. Cost Analysis

6.1 Price List

<table>
<thead>
<tr>
<th>Core components</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item</td>
<td></td>
</tr>
<tr>
<td>eBox-3310A-MSJK</td>
<td>275$</td>
</tr>
</tbody>
</table>
Zigbee Coordinator (JN5139)  To collect and transmit data deployed near the eBox.  18$  

LCD  The user interface of eBox.  43$  

Total  336$  

**Extended components**  
(the amount is determined by the scale of the farmland)  

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Price</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Solenoid Valve</td>
<td>To control the supply of water and fertilizer.</td>
<td>2$ ×4</td>
<td>These components are used in one field about five acres. So if there are more fields or a larger one, you need more extended components.</td>
</tr>
<tr>
<td>Zigbee Router(JN5139)</td>
<td>Control Valve &amp; Data Acquisition.</td>
<td>18$</td>
<td></td>
</tr>
<tr>
<td>Relay</td>
<td>A switch using low level to control high level circuit.</td>
<td>1$×4</td>
<td></td>
</tr>
</tbody>
</table>

Total  30$  

**Optional Components**  

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Price</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPRS Module</td>
<td>To transmit the data to the Internet.</td>
<td>74$</td>
<td>It’s for the upgraded edition YUN-IRF. Users have stronger purchasing power. And GPRS network is covered around that region.</td>
</tr>
<tr>
<td>mini2440</td>
<td>To convert socket to WCF.</td>
<td>40$</td>
<td></td>
</tr>
</tbody>
</table>

Total  114$  

**6.2 Analysis**  

<table>
<thead>
<tr>
<th>System</th>
<th>Farm with small scale(200~400 acre)</th>
<th>Farm with mid Scale(400~1000 acre)</th>
<th>Farm with Large Scale(1000~)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Core</td>
<td>Extende d</td>
<td>Option al</td>
</tr>
<tr>
<td>IRF</td>
<td>1</td>
<td>3~7</td>
<td>0</td>
</tr>
<tr>
<td>YUN-IRF</td>
<td>1</td>
<td>3~7</td>
<td>1</td>
</tr>
</tbody>
</table>

**7. Overall Architecture**  

Our system consists of three layers—L1, L2 and L3. By taking each layer’s computational capabilities, storage and power supply into consideration, we have to implement algorithms with different complexities in different layers so as to balance the tradeoff between hardware capability and system requirements. Therefore the computing and hardware efficiency are much improved. L1 and L2 is essential for both the IRF and YUN, while the L3 is just for the upgraded edition—YUN-IRF. The structure of the whole architecture is shown in fig below.
8. Hardware Components

8.1 Slave System

Both the lite edition—IRF and the upgraded edition—YUN-IRF both have the slave system. The structure chart is shown in the fig below.

The slave system is a real-time controlling one and the weather condition of the field will decide the system to work or sleep. The slave system will collect the current data of the specific field every two minutes, encapsulate the data into packet and then transmit the packet to the master system. What’s more, the slave system will also receive commands from the master system and then execute the command that determines the supply of the fertilizer and water flow, thus irrigating the field automatically.

**8.1.1 Data Acquisition Module**

The temperature sensor and humidity sensor is integrated with the Zigbee router deployed in the different parts of the farmland. Here is a figure in one part of the farmland, as shown in the picture behind. But it’s not practical for we cannot put the Zigbee node in the open-air. In the further, in order to separate from Zigbee node and collect the actually data, we can use other temperature and humidity sensor which can be put under the soil. While the sensor pass back analog signal, we achieve ADC processing on Zigbee router for it can relieve the computing pressure on eBox.
8.1.2 Fertilizer & Water Control Module

For each part of the farmland, IRF need 4 solenoid valves to control the supply of fertilizer and water flow separately. The solenoid valve needs 12V voltage to work and the relay needs a 3.8V voltage. However, the Zigbee node is powered by two batteries, so the high voltage level is only 2.8V, too low to drive the relay much less the valve.

So we firstly lift the 2.8V voltage to 4.4V, which is enough to drive the relay, then use the relay to act as the switch to control the 12V circuit. Through this way, we can enable or disable the relay, thus controlling the state of the valve, by controlling the node’s I/O pins to output low level (0V) or high level (2.8V).

8.1.3 Data Processing & Transmit Module

Communication between Zigbee nodes

The use of wireless sensor network can largely lower down the difficulty in configuring the network. But the following problems need to be overcome:

1. When there is more than one router in use, how to make up a network;
2. When the distance between the coordinator and the router is too long that the data cannot be transmitted directly;
3. When the user wants to add a router into the network or quit one router from the network, how to adjust the configuration of the network automatically;
4. The nodes are powered by the batteries and users should not change the batteries too often, how to save the energy.

For these, we adapt a kind of serialized peer-to-peer wireless network as a solution, which is based on the standard of IEEE 802.15.4 and adopts multi-hop network. In the solution, all nodes are numbered according to its short address, and all the short addresses can make up an arithmetic progression with the common difference 1 and the node with the short address of 0 is the coordinator. Each router can only receive data from the node with the next smaller address (upstream) or the next larger address (downstream). In order to function well, the nodes with the neighboring number can communicate with each other. When the router needs to receive or send data, the node will be waked up. After the process is finished, it will go to the sleep mode to reduce the energy consuming.

Data Processing on Zigbee routers

To be real-time, a Zigbee router will read the current data from the sensors every two minutes and encapsulate the data into a packet, then transmit the packet to the next node. The interval of two minutes is short enough to satisfy the demand of being real-time but long enough to ensure the eBox establish TCP link with the Cloud Server.

8.2 Master System

Our master system is various from different series of IRF. For lite edition—IRF aimed at low-end market, master is made up by the Zigbee coordinator, eBox and LCD screen. For the high-end market we add Cloud server to improve its function.
When to irrigate the field, how much water or fertilizer is needed should be determined by the master system. The process of determining involves many factors such as the temperature, the humidity, the farmland’s location in the world and the growth stage of the crop. It will be better that the user can check the historical records of irrigation.

8.2.1 Zigbee Coordinator

The coordinator will receive data from the router and transmit the data to the eBox through the serial port cable. After the eBox processing the data and returning a relevant command, the coordinator will transmit the command to the router wirelessly.

8.2.2 The control core—eBox

Powerful Controlling Center

As is said above, the eBox is vital in two editions. For the lite edition—IRF, the system will be controlled by eBox completely. While for the upgraded edition—YUN-IRF, the Cloud Server gets the access to the system. No matter which edition it is, the eBox will receive data from the routers and process the data. Finally the eBox will send relevant command to the specific router via the coordinator to determine the working state of the valve.

Friendly Interactive Interface

Farmers in different countries may use different languages, and their degree of education may differ greatly. In order to satisfy the demand of different farmers we attach a LCD screen to the eBox via the VGA connection and provide a easily-operated interface. The interface use pictures to explain the relevant words or different colors corresponding to relevant button. With these assistant pictures, users can easily choose the proper options in the interface.

Interaction to Cloud Platform

eBox can have data interaction with the Cloud Server through the GPRS module. The data receiving from the nodes will be processed and then transmitted to the Cloud Platform. The Cloud Database will then record the data and update. Meanwhile, if the eBox is running in assistant-mode, the length of irrigating time will be sent from the Cloud Server to the eBox.

8.2.3 YUN—for the Upgraded Edition

Our Cloud Platform is based on the Windows Azure and SQL Azure, and three important services for our Irrigation system have been created on it.

Host Services

Host service has three main parts: Data exchange, data processing and Web application.

Data exchange: To exchange data between eBox and Windows Azure, we use WCF because it can give us a great convenient way to develop a safe and useful communication interface.

Data processing: By using Windows Azure’s powerful computing ability, we can use the historical records and requirements pre-stored of different types of crop and the climate of different regions in the world. On the basis of these data, we can use Penman-Monteith & Mitscherlich-Bray equation to control the irrigation more scientific.

Web application: running an asp.net application on it.

Database

For the SQL Azure can offer us a stable database, we use it to store huge amount of previous data, type of the crop and region of the farmland for different users. And the database is the foundation of our data analysis on YUN-IRF.

Asp.net Web site

Our web is a friendly interactive interface that can allow our users to get their farmlands’ information or control their irrigation system manually. It is based on the ASP.Net and contains three parts. First, it has a register system that can identify every user. Second, our web has a query system that can allow every user to know the current temperature, humidity, and fertilizer condition and the farmlands’ history weather and climate condition. Third, it can allow the user send the command to control the irrigation system.
8.2.4 Communication Module Between Cloud Platform and eBox

Taking into consideration that it's not practical to deploy cables in the farmland, we use GPRS module to get access to Internet and add a WCF adaptor to help it to convert socket into WCF.

GPRS Module

We use Sim300, which can get access to the Internet through China Mobile's access point—cmnet.

WCF Adaptor

We use mini2440, which can run a WinCE system, as a bridge between the GPRS module and the Cloud Platform. On mini2440 we run a WCF interface that can communicate with the Cloud Platform, and a socket interface that can communicate with the GPRS module. Because mini2440 needs to communicate with the Internet, it is easy to deploy at any where that can access the Internet.

9. Overall Embedded System Architecture

9.1 Embedded Image

In order to ensure the operation on eBox-3310A, we adopt our own designed Windows CE 7.0 image with Visual Studio 2008. We newly establish a project of the type of Platform Builder to develop Compact 7 OS Design project, and choose ICOP_eBox3310A_70CTP: x86 as the BSP. The main components are as follows:

- **Console Window**: when debugging our application, the console window is essential.
- **Wired Local Area Network**: when we debug our own application on the eBox, we must establish connection with the eBox with Network cable, so this component is necessary.
- **Hive-based registry support**: when power resetting the system, it’s necessary to remain the parameters in the flash storage.
- **VGA-03 800×600×16 @ 60 Hz**: to support our LCD screen;
- **1st Serial Port, 2nd Serial Port**: in our system, the eBox should connect with the Zigbee Node and the GPRS module through the serial ports.
- **USB Auto driver, USB Mass Storage class driver**: they are necessary to support our USB storage.
- **8042 PS/2 Keyboard/Mouse, USB HID Keyboard and Mouse**: they enable the eBox support the mouse/keyboard with the USB interface or PS/2 interface.
- **CAB File Installer/Uninstaller**: It enables installer/uninstaller of application program on the eBox, what’s more, the debugging and deployment of the VS2008 application cannot go without its support.
- **.NET Compact Framework 3.5, OS Dependencies for .Net Compact Framework 3.5**: .NET environment provides a lot of encapsulated class for application software development, since the main development language on the eBox is C#, thus this component is required.
- **FTP Server**: it provides convenience for users to upload files to the eBox or download file files from the eBox.
- **CoreCon_v100_Compact 7**: to enable the eBox to establish connection with the VS2008 workstation, so that we can debug our application online.
- **AutoLaunch_v100_x86_Compact 7**: to enable the eBox to run our application when starting.
- **Remote Display Application**: it provides the possibility of accessing the desktop remotely, which make a big convenience for us debugging the application.

Power On Start

As an embedded system, it should run the system automatically. So we change the register to achieve it. Meanwhile we make the image that including our application.
9.2 Embedded Software Design
Embedded software includes three parts:

**eBox**

We use VS2008 to develop the application, using the C#. This part mainly realizes the following functions. One is to accomplish communication with the Zigbee nodes. The eBox receive data from the Zigbee node, after processing, relevant commands should be sent back to the node again. The second function is to accomplish interaction with the users. When users firstly use the device, he must set the surrounding that the system works at. The interface is simple and anyone who knows the appearance of relevant crops and can recognize the colors can operate the device correctly. The last one is to control the system when it’s used in the IRF. In the IRF, there is no Cloud platform and the eBox will control the system alone. In this situation, the eBox will run the Experienced Algorithm to determine the proportion and amount of irrigation.

**Zigbee nodes**

The next part is working on the Zigbee nodes, which is developed using the C language. As there are two kinds of nodes, which are the routers and coordinator, the software also can be divided into two kinds. The application working on the router is responsible for collecting data and then transmitting them to the coordinator. Furthermore, the application can handle data receiving from other nodes, if the data is a command, the application must work to control the IO pin to output high or low level, which will control the valve. The application working on the coordinator is designed to receive the data from the node or the eBox, and transmit it to the eBox or the node.

**Cloud Platform**

The last part is working on the Cloud Platform. Our application on the Server contains four parts. The first part is the WCF Interface which transfers data from WCF adaptor. The second part is the Compute module that gets data from the SQL module, then calculates those data by using Penman-Monteith & Mitscherlich-Bray equation and transmits the result back to the WCF Interface. The third part is the Web interface which is based on the ASP.NET, we use div+css to make the web page more beautiful. The last part is the SQL module, which is to build a database on Cloud and operate it.

**WCF Adaptor**

For our GPRS module doesn’t have a fixed IP address that allows the server to send the data initiative, we designed the application on the mini2440 for three parts: WCF Interface, Socket Interface, using the C# language. The function of WCF interface is to communicate with our Windows Azure application, and the function of Socket interface is to communicate with our eBox program. We save commands transferred from the server and send it to the GPRS module when the TCP connection has been created by Socket protocol.

9.3 Why not PC

As is known to all, the embedded device has higher integration, smaller size, lower power consumption and other advantages, which make it more suitable for doing some special things.

When our system is to put into production, the stability is of great importance, because the device may be running for months without management. The eBox can satisfy the demand with his specific function.

The control must be real-time, so only an embedded device which has a rapid root and relatively single function meets the requirement. What's more, the device should run all the time, so the consumption should be taken into consideration. The following table is the comparison between a PC and the eBox.

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>Ebox</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amount</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Power consumption</td>
<td>200 W</td>
<td>10 W</td>
</tr>
<tr>
<td>Stability</td>
<td>Relatively low</td>
<td>High</td>
</tr>
<tr>
<td>Recovering time</td>
<td>Long</td>
<td>Short</td>
</tr>
<tr>
<td>Size</td>
<td>18.5<em>38.5</em>41.6 cm</td>
<td>11.5<em>11.5</em>3.5 cm</td>
</tr>
<tr>
<td>Weight</td>
<td>&gt;7000 g</td>
<td>505 g</td>
</tr>
</tbody>
</table>
In conclusion, taking factors in the fig above into consideration, the EBox is the best choice for users.

10. Data Flow

10.1 Data Transmission and Protocol

<table>
<thead>
<tr>
<th>Transmission</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>Between Zigbee node</td>
<td>IEEE 802.15.4</td>
</tr>
<tr>
<td>Zigbee module &amp; ebox</td>
<td>RS232-DB9 Serial Port Protocol</td>
</tr>
<tr>
<td>Ebox &amp; GPRS module</td>
<td>Rs232-DB9 Serial Port Protocol</td>
</tr>
<tr>
<td>GPRS module &amp; WCF Adaptor</td>
<td>Socket /Tcp protocol</td>
</tr>
<tr>
<td>WCF Adaptor &amp; Cloud Platform</td>
<td>WCF/Tcp protocol</td>
</tr>
</tbody>
</table>

10.2 The Data Flow of YUN-IRF and Analysis

Taking the actual situation into account, we define two working modes. The one is YUN-IRF; the other is IRF. The data flow of YUN-IRF is shown in the following figure.

Zigbee Router collects the data from the sensors, which are integrated with Router in the same PCB board, and transmits them to Zigbee Coordinator through local wireless network. The eBox gets the data from Zigbee Coordinator via the serial port, then intercepts the address number of the Router and then transfers the data to the Cloud platform though the GPRS Module. Because the Cloud Platform doesn’t support socket but WCF, an adaptor is needed. After the TCP connections between GPRS Module & WCF Adaptor, WCF Adaptor & the Cloud Server have been set up successfully, GPRS Module submits the data to the Cloud Server and waits for callback. Before the Cloud Server accomplishes the complex calculation about the detail irrigation information, the Cloud Server stores the data to the Cloud Database which can be viewed through web page. GPRS Module gets the callback through the network and transmits the irrigation information to the eBox. Then the eBox analyzes the data and sends open or close command to the specific Router with help of the Coordinator to complete one turn of irrigation.

The data flow of IRF is shown as follow figure.

The characteristic of IRF is that the information about the irrigation is calculated by the eBox in a relative simple way or even can be controlled manually. The most important points are that its network is localized so it can be applied to the remote area with no signal and without GPRS Module and Cloud peasantry can save money for the relevant hardware.

10.3 Data Format

10.3.1 Zigbee Router to Coordinator & Coordinator to eBox

Let’s take an example of data which are shown in string.

“Address = 1\nHumidity = 40\nTemperature = 27\nVoltage = 2910”
“Address” implies the address number of the router in the local network. “Humidity” and “Temperature” is the core of the data. “Voltage” is the data from which the system can tell whether the voltage is enough to keep the Router running.

10.3.2 eBox to GPRS Module & GPRS Module to Cloud

The eBox adds a header to the data it received and transfers it to the GPRS Module. The content of the header is determined by the choices about crop category, climate type and grow stage of the crop the farmer made. One possible header in string may like this:

“category = corn
climate = Arid
growstage = Initial”

10.3.3 Cloud to GPRS Module & GPRS Module to eBox

After Cloud finishes the calculation, it transmits the time length of opening the valve to the eBox. Here is a sample in string of it.

“water=600000 fertilizer1=360000 fertilizer2=240000 fertilizer3=120000”

The numbers indicate the length of the time a specific solenoid valve should open.

**Note: All the numbers are in milliseconds.**

10.3.4 eBox to Zigbee Coordinator & Coordinator to Router

The eBox sends open or close command to the specific Router through Coordinator and the interval of these two commands is controlled by the eBox according to the irrigation amount data.

We can take a look at a command sample which is in hexadecimal: 0x00010201.

The first and second bytes (0x0001) represent the address number of the Router. The third byte (0x02) indicates the number of solenoid valve. The last byte implies whether to open or close the valve: 0x00 means open while 0x01 means close.

11. Data Processing

The core of the data processing in Cloud is the calculation of the time length of irrigation especially the time length of irrigating water. The time length can be computed by the amount of water deserved to be irrigated. The amount of water is estimated by the crop evapotranspiration ($ET_c$). $ET_c$ is determined by crop coefficient ($K_c$) and the reference evapotranspiration ($ET_0$).

$ET_0$ can be calculated according to the Penman-Monteith equation which is recommended by the Food and Agriculture Organization of United Nations (FAO).

$$ET_0 = \frac{0.408\Delta(R_n - G) + \gamma \frac{900}{T + 273}u_2(e_s - e_a)}{\Delta + \gamma(1 + 0.34u_2)}$$

Where

- $R_n$ = net radiation at the crop surface
- $G$ = soil heat flux density
- $e_s$ = saturation vapour pressure
- $e_a$ = actual vapour pressure
- $\Delta$ = the tangent slope of the saturated vapor pressure-temperature curve at the point of temperature equals T
- $T$ = mean daily air temperature at 2 m height
- $\gamma$ = Humidity

$K_c$ can be adjusted for climate according to the equation as follows:

$$K_c = K_c(\text{standard}) + [0.04(u_2 - 2) - 0.004(RH - 45)]\sqrt{h}^{0.3}$$

Where

- $K_c(\text{standard})$ = standard crop coefficient
- $h$ = mean plant height
- $RH$ = mean daily minimum relative humidity

The equation of $ET_c$ is as follows:
\[ ET_c = K_c \cdot ET_0 \]

Steps for Cloud to calculate the amount of the water are as follows:

1. Calculate the coefficient T and RH basing on the previous and current data.
2. Search the necessary coefficients of the above two equations from the database according to the crop category, climate type and grow stage.
3. Substitute the value of every searched coefficients, the data of current temperature and humidity into the equation to calculate \( K_c \) and \( ET_0 \).
4. Calculate \( ET_0 \) by multiple \( K_c \) and \( ET_0 \).
5. The amount of water equals \( ET_0 \) minus the mathematical expectation of the precipitation.
6. Convert the amount of water to the length of solenoid valve opening time.

The result of computer simulation is shown in the Fig:

According to the Soil Fertility Growth Response Curves, we can determine the amount of every fertilizer. Mitscherlich-Bray model is the most popular one, the equation is as follows:

\[
\log(A - Y) = \log A - c_1 B - c X
\]

Where
- \( A \): theoretical maximum possible yield
- \( Y \): actual yield
- \( X \): amount of nutrient added
- \( c_1 \): const, efficiency of soil
- \( c \): const, efficiency of fertilizer nutrients
- \( B \): native soil test value

The characteristic curve of the equation is shown in Fig:

Steps for Cloud to calculate these parameters are as follows:

1. Check whether the length of time from last fertilization till current time is long enough. If not, return 0.
2. If it is long enough, search the value of every coefficient of different crops and different grow stages from the Cloud Database and calculate the amount of nutrient should be added.
3. Convert the amount of the nutrient to the amount of the liquid fertilizer whose concentration exactly equals the concentration of cytosol.
4. Convert the amount of the liquid fertilizer to the length of solenoid valve opening time.

The above data processing is based on Cloud Platform. If eBox cannot connect to the Cloud Platform, eBox does the processing work in a relative simple way.

The amount of water can be calculated as follows:

\[
W = P (H_o - H_u) \cdot C \cdot D
\]

Where
- \( W \): amount of water
- \( P \): const, varies in the different crops
- \( H_o \): humidity
- \( C \): soil bulk density
Ho - water-holding capacity of the field  
D - the depth of the soil that impact the crop

The amount of the fertilizer is similar with Cloud Platform but it’s not necessary to search the value of coefficients and calculate in step (2). All fertilizer with fixed amount will be provided at the same time.

12. Testing Procedures

12.1 Hardware Test

The PCB board with Zigbee Module and sensor, eBox, GPRS Module and solenoid valve circuit all need hardware test.

12.1.1 PCB board with Zigbee Module and sensor

The two LEDs on the PCB board can indicate the working state of the Zigbee Module and the sensors. When the module works normally, the first LED will always be on. The meaning of the other LED varies from the different roles in the local network. To the Router, the fact that the second LED is on and off periodically means that the Router have sent data successfully. To the Coordinator, it means the local network has been set up triumphantly.

12.1.2 eBox

To test eBox, we first install a screen with VGA interface. Then we can test the two serial ports with the help of the serial port assistant program under the circumstance of WINCE 7.

12.1.3 GPRS Module

The frequency that the indicator light twinkles shows the working situation of the GPRS Module. After power on, the frequency of the indicator light twinkles is high, which implies the GPRS Module hasn’t finished the initialization yet. After a few seconds, the frequency of the indicator light twinkles becomes low. That is to say, we can send commands naturally.

12.1.4 Relay Circuit

Using a multimeter, we measure the output of the voltage-lifting circuit. When the input is zero, the output is 0.45 V; while the input is 2.8 V, the output is 4.5 V, which is enough to drive the relay.

12.2 Software Test

12.2.1 Zigbee Module

On the one hand, we collect the data Router sends and check it via serial port assistant. From the figure behind we can see the Zigbee router is now transmitting data to the other node. On the other hand we transfer open and close command and check the second LED of Router.

12.2.2 eBox

We disconnect the Zigbee node with the eBox and give a const value to the variable that accepts the data the coordinator received. What’s more, in order to test the mode with Internet, we replace the Cloud Server
with a TCP server program on PC while debugging. As for the mode without Internet, we just assume the data in string from coordinator only. As is show in the figure above, we can see the GPRS Module is feedback—“CONNECT OK”, which illustrates our eBox have connected to the WCF adaptor.

12.2.3 Host services & SQL Azure test

From the fig below we can see that our host services & database works well.

12.3 System Test

Because no bug is found in separate tests above, we integrate all the modules and try a system test. No assuming data or any replacement is permitted and we test several turns of data flows in different modes. The outcome justifies that there is no severe problem.

13. Performance Analysis

Based on the above tests, we figure out several vital indicators of the whole system.

13.1 Response time Test

We test all the response time of the software systems on Zigbee Module, eBox and Cloud Platform. Although the response time of software system on Cloud Platform may be somewhat disappointing in multi-users, the response time of Zigbee Module and eBox is short.

13.2 Robustness Test

We also test the robustness of the software system. The result of Zigbee Module and eBox is perfect. The possibility that the Cloud Platform get an answer which deviate from the correct value is tiny and the error is not so big that can be ignored.

13.3 Endurance Test

We use two 1.5V batteries to supply Zigbee module and eight 1.5V batteries to supply solenoid valve circuit. Zigbee module works more than X hours continuously in the awaking situation and meanwhile the solenoid valve circuit works more than X hours continuously in the awaking situation.

13.4 Stability Test & Multi-User Test

The outcome of the Cloud Platform is favorable, but the wires connect solenoid valve and circuit is not very stable because of the wind outdoor. Taking the advantages of Cloud Platform, the number of users may be large.

14. Project Status