# 概述

Node是基于Arduino的物联网节点平台。该平台配合SeeedStudio的Grove电子模块系列和Xbee等通讯模块，用户可以快速地组建智能家居网络，并实现一定逻辑控制。使用Node平台来实现一个智能家居系统，例如远程人体监测报警、远程测距等，将变得非常方便和容易。

Node分为Atom.Node和Cloud.Node。

Atom主要用于局域网的通信。Atom通过Xbee或RFbee等无线模块进行相互通信，通信可靠，通讯距离高达100m；在配置上，用户使用SeeedStudio公司开发的基于Android平台的App，使用可见光通信的方式对Atom进行配置，配置过程简单直观；在使用上，Atom可以被单独使用来实现简单的IFTTT功能，也可以配对使用以实现远程无线控制和监控功能。

Cloud配备Xbee接口以及Wi-Fi功能，所以Cloud除了可以与Atom通信之外，也可以与互联网进行通信。配合Cloud，用户可以很方便的将Atom采集到的信号数据等上传至云端，或从云端对Atom进行简单控制。这将是一个很方便易用的智能物联网解决方案。

## 整体架构

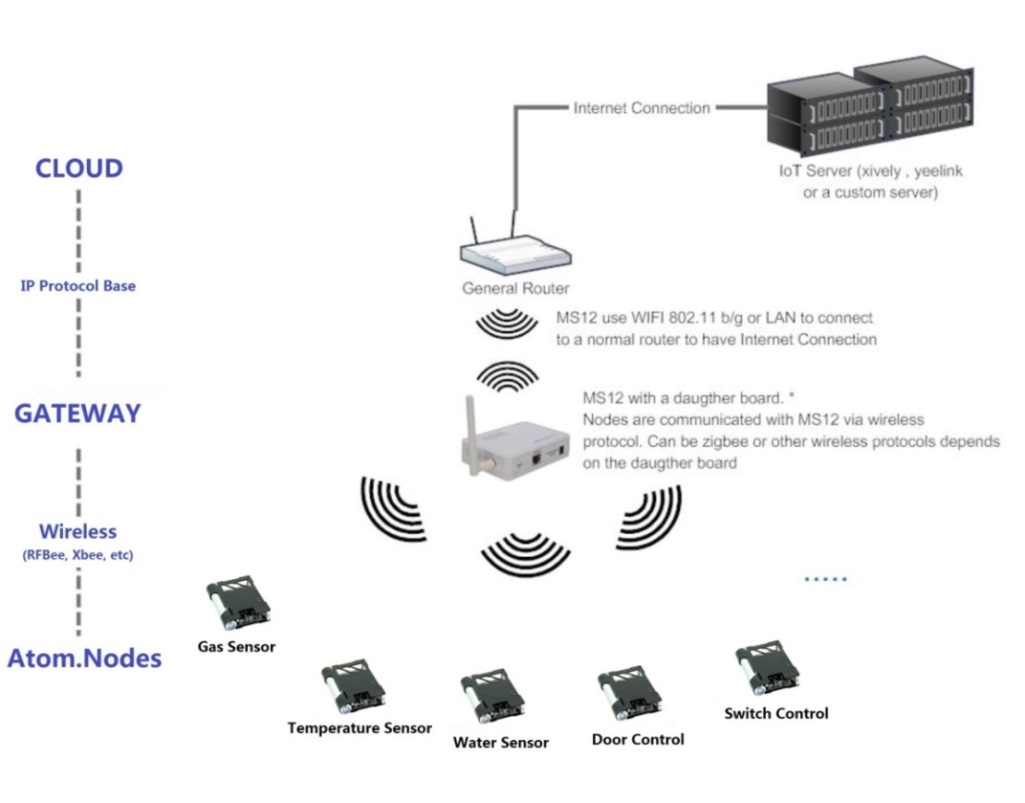


图 1‑1 Node工作示意图

上图反映了Cloud.Node是如何工作的，该架构有三个主要的部分：

* **IoT Server:**存放传感器数据的服务器，它们同时可以向你的设备发送一些命令，一些公共的服务器例如[www.yeelink.com](http://www.yeelink.com)， [www.xively.com](http://www.xively.com)
* **Gateway**: 传感器并不能直接跟IoT Server通信，需要通过一个网关。Cloud.Node就是扮演这个角色，当Cloud.Node接收到传感器节点的数据后，通过TCP/IP协议把数据发送到IoT Server
* **Nodes:** Atom.Node，各种各样的Sensor或Actuator

## 硬件组成

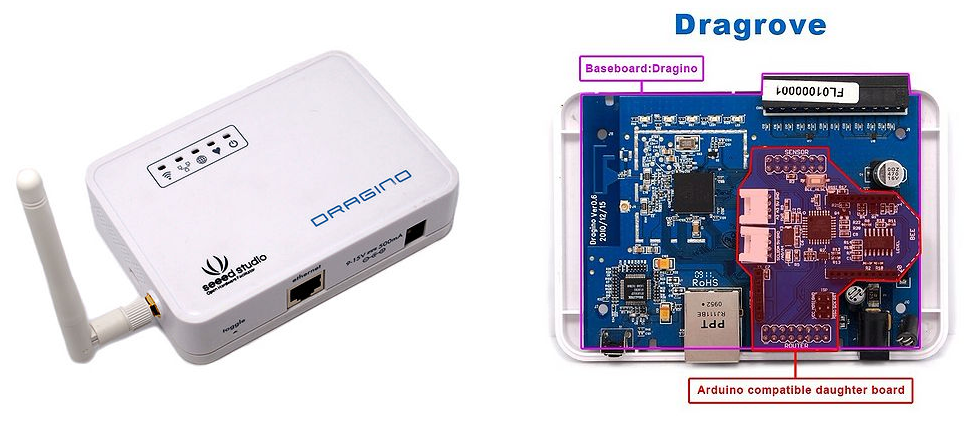


图 1‑2 Cloud硬件组成

Cloud的硬件由两部分构成，Dragino以及插在上面的daughter board。Dragino与daughter board通过Uart以及SPI通信。

Daughter board上留有Xbee的接口，可以接Xbee，RFBee等通信模块与Atom.Node通信。Daughter board负责管理Node网络各Atom的管理，把数据通过Dragino与云端联系。

Dragino是一块基于OpenWrt的主板。

# Daughter board通信接口

Daughter board通过串口向Dragino发送命令，为了便于管理和扩展，定义了以下的通信格式：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| START | DATATYPE | SPACE | DATA | END |

**START**: “ss”

**DATATYPE**: 1: ADD NODE; 2: POST DATA; 3: DELETE NODE

**SPACE**: a space between DATATYPE and DATA

**DATA**: Valid Data

**END**: “gg”

## 命令

### 添加一个设备

当有一个新的Atom.Node加入网络时，需要添加一个设备，发送以下命令可以完成这个功能。

**Format**: “ss1 NODE\_ID, SENSOR\_ID, ACTUATOR\_IDgg”

**Function**: Create a Node with Specify SENSOR\_ID and ACTUATOR\_ID

**Example**: “ss 1 45,2,201gg”

上面的Example创建了一个device\_Id为45，sensor\_Id为2，actuator\_Id为201的设备。

Once MS12 get this command from its UART port, MS12 will:

当Dragino接收到以上的命令后，会添加一个设备到设备管理器中，可以通过网页配置端的Devices List看到：



### 推送数据

**Format**: “ss2 DEVICE\_ID,VALUEgg”

**Function**: 推送一个传感器数值到 IoT server

**Example**: “ss2 45,30gg”

### 删除一个设备

**Format**: “ss3 DEVICE\_IDgg”

**Function**: 删除一个设备

**Example**: “ss3 45gg”

删除device\_id为45的设备

# Daughter board的程序

Daughter board的程序主要负责：

* 接收Node网络的RF数据
* 管理Node网络各节点的数据
* 发送命令进行添加节点，数据推送及删除节点

## 软件流程图

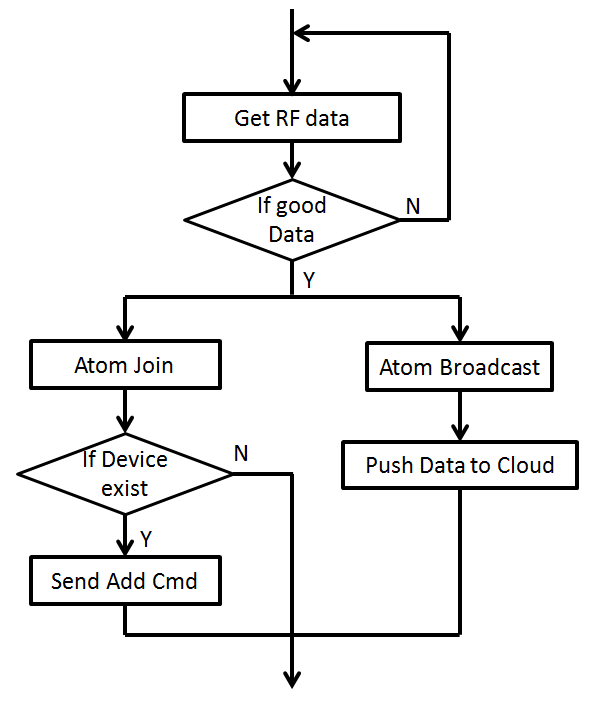


图 3‑1 程序流程图

## 主要函数接口

### 添加一个设备

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\*\* Function name: postDta to yeelink

\*\* Descriptions: return 1: ok 0: nok

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void NodeManage::yeelinkAdd(unsigned char idNode, unsigned char idSensor, unsigned char idActuator);

### 数据推送

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* Function name: postDta to yeelink

\*\* Descriptions: return 1: ok 0: nok

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

void NodeManage::yeelinkPost(unsigned char idNode, unsigned int psDta);

# 如何访问Dragino

## 通过网页配置端

网页输入192.168.255.1，即可登录到Dragino的网页配置端。



图 ‑1 Dragino的网页配置

## 通过SSH

Dragino上跑了一个嵌入式Linux操作系统:OpenWrt。可以通过SSH登录：

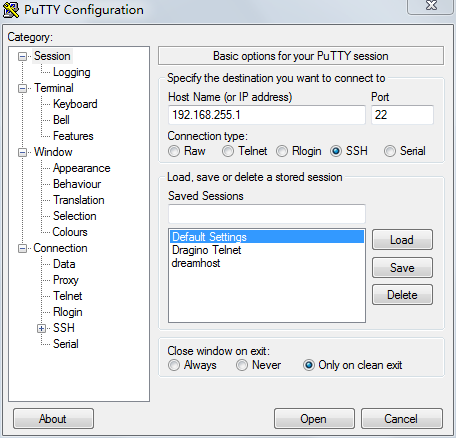


图 4‑2 通过SSH登录Dragino

The SSH access for Dragino is:

IP address: 172.31.255.254 on LAN port.

Username: root

Password: root

SSH access password can be changed by type below commands:

*root@dragino-751aff:~# passwd*

*Changing password for root*

*New password:*

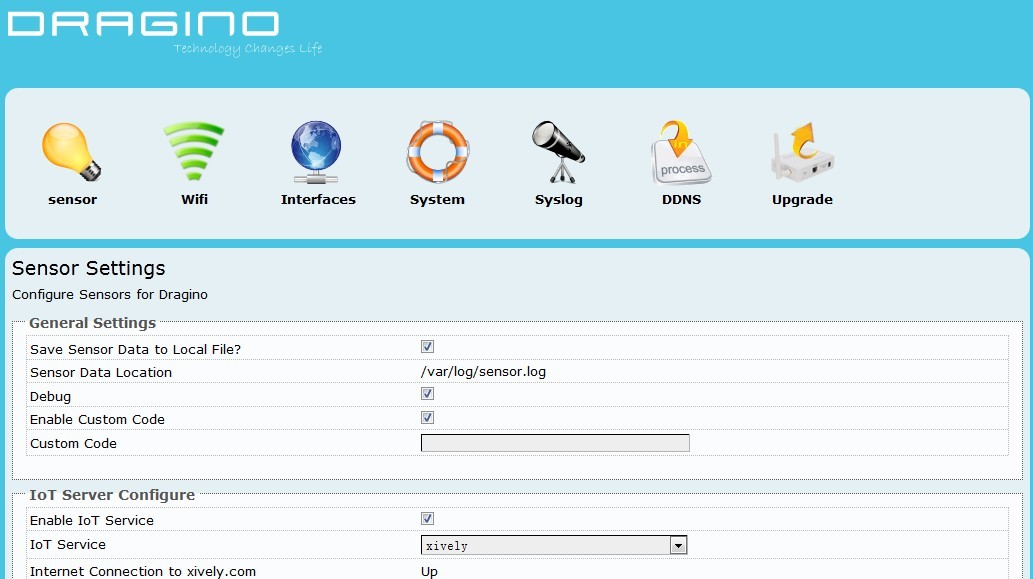
*Retype password:*

*Password for root changed by root*

**Notice**: for security reason, it is recommend that you change the SSH access password after first log in.

# 网页配置界面

## Sensor Settings



**General Settings**:

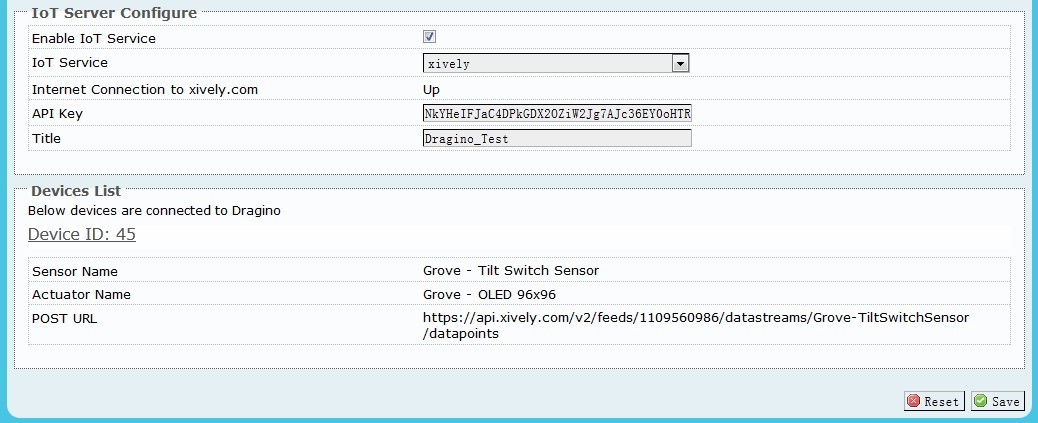
* **Save Sensor Data to Local File**: Enable/Disable local logger for sensor data from UART port**.**
* **Sensor Data Location**: File to store the sensor data. Can be changed in /etc/config/sensor
* **Debug**: Enable/Disable Debug. Debug info will show on Syslog page
* **Enable Custom Code**: Enable/Disable Custom Code.
* **Custom Code**: use Linux commands to process sensor data. The Custom commands will be executed once there is new incoming string from UART port. Developer can use macro **[RAW\_DATA]** to get the incoming string

**Example:**

echo [RAW\_DATA] >> tem.log; atftp -p -r tem.log 192.168.1.2; rm tem.log

This command will forward the RAW\_DATA to tftp server with file name tem.log

Note: Due to security reason, Enable Custom Code and Custom Code options are set to invisible by default. They can be set to visible by set the option “**show\_custom\_code**” to “1” in file /etc/config/sensor



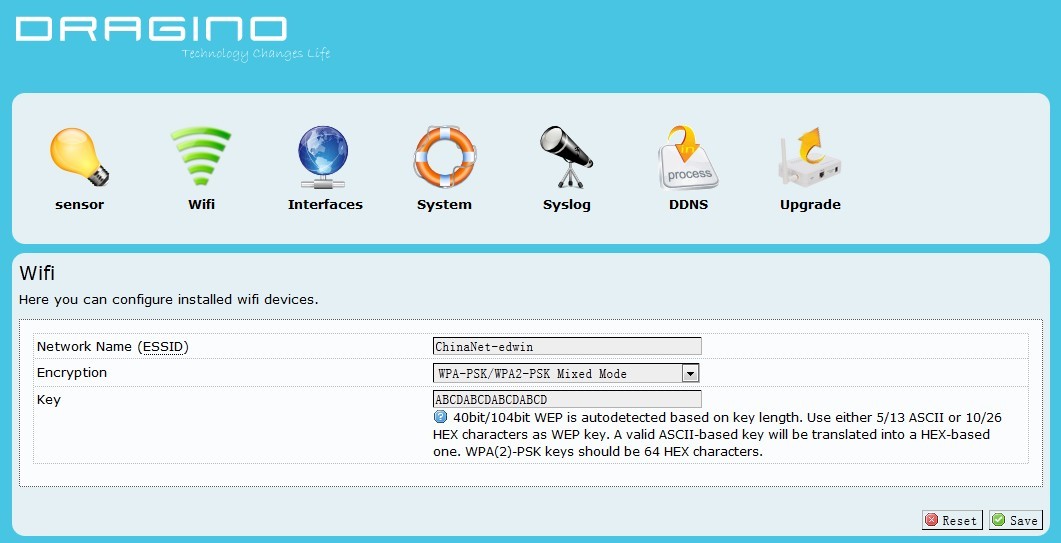
**IoT Server Configure:**

* **Enable IoT Service:** Disable/Enable IoT Service
* **IoT Service:** Choose Service Provider
* **Internet Connection:** Show the connection to the service provider
* **API Key:** API key from service provider
* **Title:** Default title prefix when create nodes on the IoT server. If leave this blank, the default tile will be set to host name of the MS12.

**Devices List:**

Shows the Node Connected to Dragino MS12.

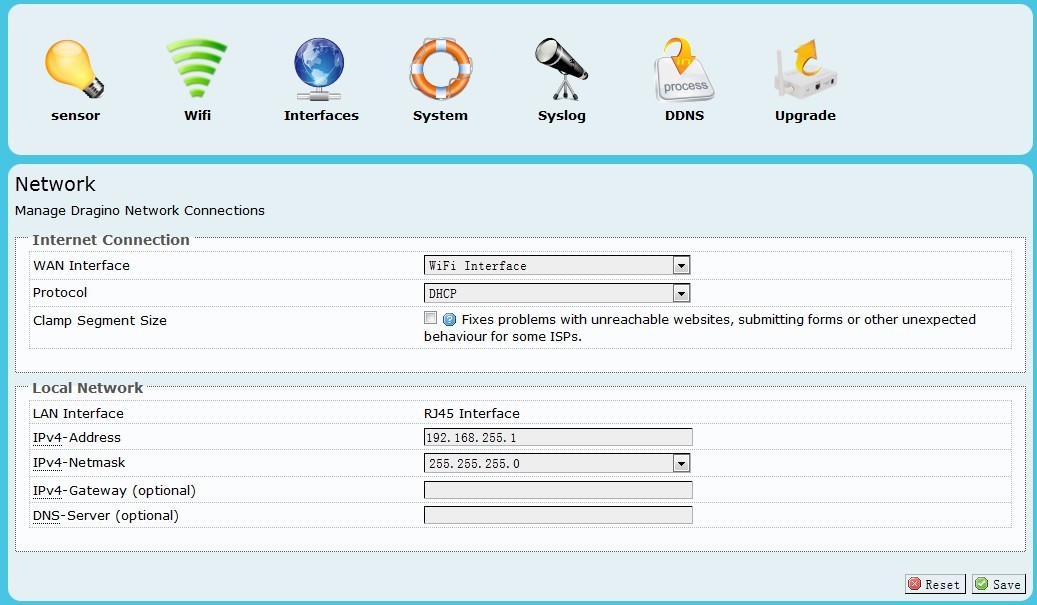
## Wi-Fi Settings

****

**Wi-Fi Settings:**

* **Network Name(ESSID):** Input your wifi router SSID
* **Encryption:** Encryption method used by your router
* **Key:** Encryption Key of your wifi network

## Network Interfaces

****

**Internet Connection**

**WAN Interface:** Interface for internet access, either WIFI or RJ45 port

**Protocol:** Way to get IP: Manual, DHCP, PPPoE

**Clamp Segment Size**: Normally don’t use this

**Local Network – Network distributed by MS12**

**LAN interface:** Interface for local network, either WIFI or RJ45 port

**IPv4-Address:** IP address for the interface used for local network.

**IPv4-Netmask:** Net mask for this local network

**IPv4-Gateway:** Gateway for this local network, default is the IP address set up above.

**DNS-Server:** DNS for this local network.

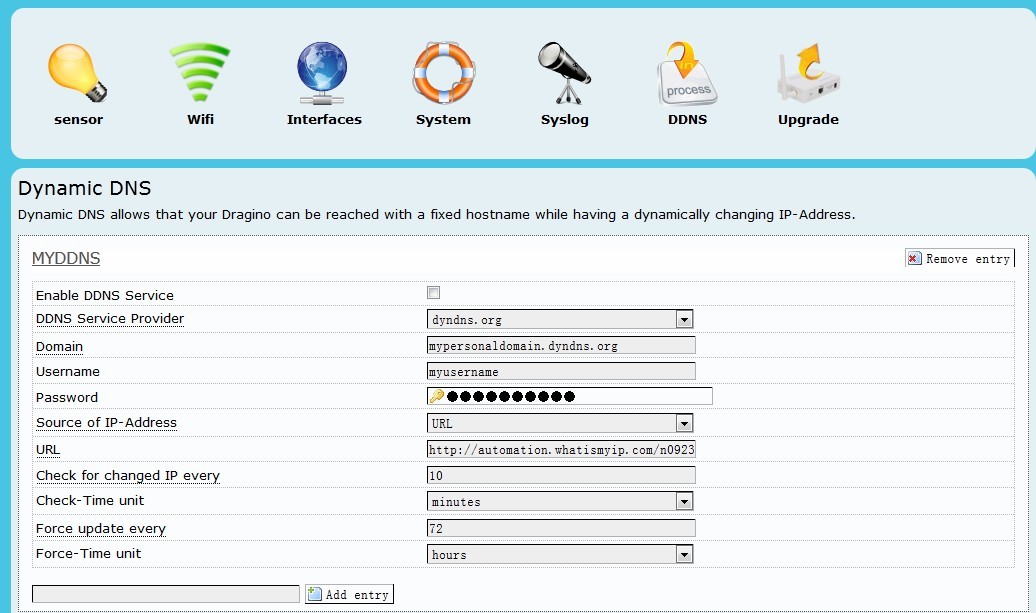
## System Status

****

System status page.

## DDNS settings

Dynamic DNS allows you access/control the Dragino & Dragrove from other location even you don’t have a fix IP Address.



***Enable DDNS Service***: Enable/Disable DDNS service

***DDNS Service Provider***: choose your service provide here

***Domain***: the hostname provide by your DDNS service provider.

***Username***: Username of your DDNS service

***Password***: Password of your DDNS service

***Source of IP-Address***: Where to look for your external IP address. You can choose:

**Network**: Set external IP according to your network interface info, e.g. wan, lan

**Interface**: Set external IP according to your hardware network interface info: eth0 , ath0.

**URL**: Set external IP according to URL info, for example, you can set it to <http://www.whatismyip.com/automation/n09230945.asp> so the Dragino will connect to this url and get its external IP. It is used when Dragino have a private IP address in its wan port.

***Check for changed IP***: how often to check if it needs to update its IP to ddns service provider.

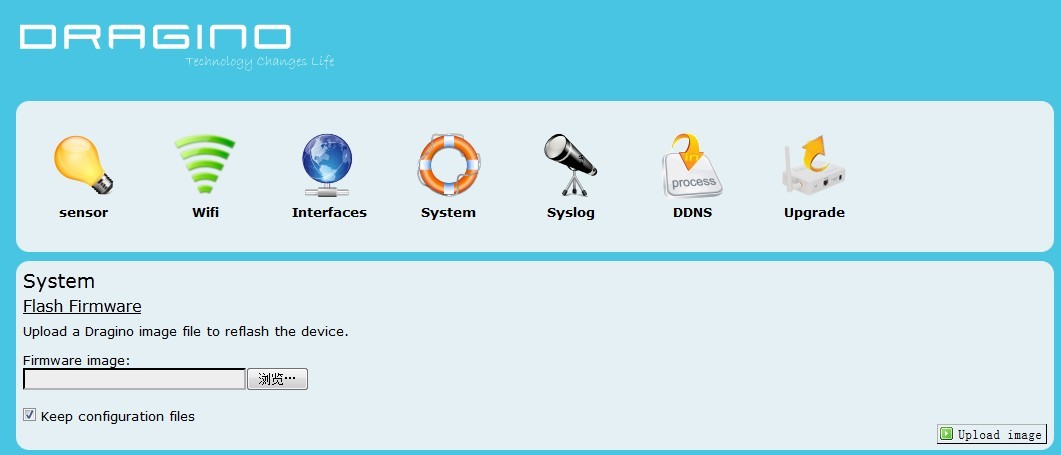
***Force update***: how often to force an update to DDNS service provider

## Upgrade Firmware

The latest firmware of Dragino can be found in below link:

<http://www.dragino.com/downloads/index.php?dir=MS12/firmware/>

Valid upgradable firmware via GUI has a suffix combined.img. For IoT application, choose the firmware in under IoT directory.



1. REFERENCE

* [www.seeedstudio.com](http://www.seeedstudio.com) : Dragrove vendor, more info about Dragrove and its development kit can be found here.
* [www.openwrt.org](http://www.openwrt.org): Embedded linux used in Dragino.
* [wiki.dragino.com](http://wiki.dragino.com/index.php?title=Main_Page): General software/hardware design info for Dragino MS12
* [www.yeelink.com](http://www.yeelink.com): A public IoT RESTful server used in China.
* <https://github.com/reeedstudio/Cloud_Dragrove>: Daughter board firmware code
* <https://github.com/reeedstudio/Cloud_Dragino_Firmware>： Dragino Firmware