Seeeduino Cloud – Parte 1



<u>Introducción</u>

Recientemente he adquirido un Seeeduino Cloud. Un clon compatible con Yun. La verdad es que es muy versátil, fácil de configurar y potente. En este primer post hablaremos un poco de él, lo configuraremos y habilitaremos su Api Rest para encender y apagar el led que incorpora la placa.

- Configuración de red
- Configuración con WebGui
- Programa de ejemplo y test

Configuración de red

Cuando arranquemos por primera vez nuestro Seeeduino, este levantará una wifi llamada **SeeeduinoCloud-AXXXX** a la que nos podremos conectar. Una vez echo esto, podremos acceder a la configuración web a través de la ip **192.168.240.1**.

	Welcome to your Seeeduino Cloud. Please enter password to access the web control panel
SEEEDUINO CLOUD	PASSWORD
	Please be sure you have cookies enabled before proceeding.
	LOG IN

El password por defecto es seeeduino.

Configuración con WebGui

Después de entrar en la web, la interfaz nos mostrará el estatus de las redes WiFi/ Eth. En la parte superior derecha encontraremos las siguientes opciones:

- SYSTEM -> Configuración global
- SENSORS -> configuración del servidor IoT
- UPGRADE -> Actualización de firmware

	WELCOME TO SEEEDUINO CLOUD	SYSTEM
		SENSORS
		UPGRADE
CLOOD		

Seleccionaremos SYSTEM y configuraremos nuestro Seeeduino Cloud:

- Primeramente podemos modificar el password de acceso
- A continuación seleccionaremos la red wifi a la que queremos conectar nuestro Seeeduino Cloud y estableceremos el password de la misma
- Finalmente, podemos proteger nuestra red con una password para la Api Rest. Si lo hacemos debemos tener

en cuenta que será una protección Basic Atuh y el usuario por defecto será root

For more advanced network configuration features, see the <u>advanced configuration panel</u> (luci)			
CLOUD CONFIGURATION SEEEDUINO CLOUD NAME * Seeed PASSWORD CONFIRM PASSWORD			
TIMEZONE * Rest of the World (UTC)			
DISCARD CONFIGURE & RESTART REST API ACCESS REST API ACCESS Image: Open Open Open Open Open Open Open Open			

Cuando pulsemos en **CONFIGURE & RESTART** nuestro Seeeduino se configurará, se reiniciará y se conectará a nuestra red.

Bien lo siguiente sera subir un código de ejemplo al Arduino para poder testear la aplicación. Para más información sobre como subir un programa des del IDE a nuestro Arduino podéis mirar este post <u>Arduino – Primera Parte</u>. Utilizaremos un ejemplo de la librería **Bridge**.

◎ sketch_jun29a Ard	uino 1.6.9				_	×
Archivo Editar Progra Nuevo Abrir Abrir Reciente Provecto	ma Herramientas Ctrl+N Ctrl+O >	Ayuda				₽ ▼
Ejemplos Cerrar Salvar Guardar Como Configurar Página Imprimir Preferencias Salir	Ctrl+W Ctrl+S Ctrl+Mayús+S a Ctrl+Mayús+P Ctrl+P Ctrl+Coma Ctrl+Q	Ejemplos Construidos 01.Basics 02.Digital 03.Analog 04.Communication 05.Control 06.Sensors 07.Display 08.Strings 09.USB 10.StarterKit_BasicKit 11.ArduinoISP	> > > > > > > > > > > > > > > > > > >			
		Bridge EEPROM Firmata SoftwareSerial SPI Temboo Wire RETIRADO Ejemplos de Liberías Personalizadas Ethernet GSM LiquidCrystal SD Servo Stepper TFT WiFi		Bridge ConsoleAsciiTable ConsolePixel ConsoleRead Datalogger FileWriteScript HttpClient HttpClientConsole MailboxReadMessage Process RemoteDueBlink ShellCommands SpacebrewYun TemperatureWebPanel TimeCheck WiFiStatus YunFirstConfig YunSerialTerminal		×

Una vez subido nuestro programa atacaremos la Api Rest para encender y apagar el led que vine incorporado en la placa, exactamente el 13 (situado justo detrás del puerto microusb). En esta caso utilizo el plugin para Firefox <u>RESTclient</u>, aunque podríamos utilizar cualquier otro. En la url especificamos la ip de nuestro Seeeduino, arduino(podemos acceder a otros elementos de la placa), el tipo de pin (en este caso digital), el pin (13) y el valor (0 apagado, 1 encendido). La url final sería http://192.168.1.36/arduino/digital/13/1 (encendido) o http://192.168.1.36/arduino/digital/13/0 (apagado)

[-] Request	
Method GET VRL http://192.168.1.36/arduino/digital/13/1	D
Body	
Request Body	.:
[-] Response Response Headers Response Body (Raw) Response Body (Highlight) Response Body (Preview)	
Pin D13 set to 1	

<u>Conclusión</u>

Como podemos ver la potencia de Seeeduino Cloud es mucha. Fácilmente podemos acceder a nuestro Arduino recibiendo y enviando valores a través de una sencilla Rest Api. En el próximo post acoplaremos un Relay Shield, esto nos permitirá controlar remotamente aparatos electrónicos (dado el calor que hace ahora, un pequeño ventilador)