

Domótica con Arduino



souliss

Distributed Home Automation and Internet of Things

ÍNDICE

1	Introducción.	2
2	Conexión de sensores e actuadores.	3
2.1	Tipología de E/S.	3
2.2	Sensores.	4
2.3	Actuadores.	6
2.4	Interfaces HMI.	10
3	Comunicaciones.	12
3.1	Comunicaciones serie asíncronas.	12
3.2	Comunicaciones Ethernet y Wi-Fi.	14
3.3	Bus I2C.	16
3.4	ZigBee.	17
3.5	Comunicaciones GSM.	19
4	Integración en sistemas domóticos.	20
4.1	Arquitecturas de comunicación.	20
4.2	Software de gestión: Souliss	22
4.3	Aplicación Souliss para el sistema operativo Android.	24
5	Bibliografía.	27

1 Introducción.

No presente traballo preténdese realizar unha introdución das posibilidades do sistema Arduino para abordar os temas de control e domótica nos ciclos formativos de formación profesional da familia de electricidade e electrónica.

En comparación cos sistemas domóticos comerciais, presenta a desvantaxe de non dispoñer de módulos preparados para instalar nunha vivenda directamente. Polo tanto, sería preciso en moitos casos implementar os interfaces para a conexión cos actuadores reais (persianas, electroválvulas, climatizadores, etc.). Isto pode aproveitarse para fomentar a creatividade de profesores e alumnos, permitindo afondar nos coñecementos de electrónica e programación.

As prácticas proxectadas permiten a adquisición de competencias cunha curva de aprendizaxe relativamente rápida, partindo duns coñecementos mínimos de electrónica e programación. Aínda que son casos prácticos sinxelos, poden ampliarse doadamente mediante a combinación destas para formar sistemas complexos.

Ao ser un sistema libre, están a empezar a xurdir no mercado equipos baseados en Arduino con caixas e interfaces normalizados para instalacións eléctricas estándar (carril DIN, saídas a relé, entradas opto acopladas, etc.). En consecuencia, prevese que poidan utilizarse para realizar instalacións reais cotiás a curto prazo. Por todas estas razóns, conclúese que o potencial de Arduino pode facelo útil para complementar o ensino dos sistemas comerciais.

Existen distintas placas de Arduino, con distintas prestacións segundo o microcontrolador incorporado.

Versión	Procesador	Memoria Flash/SRAM (KB)	Nº pins dixitais Ordinarias/PWM	Nº pins analóxicos E / S
Diecimila/Uno	ATmega328 (16 MHz)	32/2	14/6	6/0
LilyPad	ATmega168V (8 MHz)	16/1	14/6	6/0
Mega	ATmega2560 (16 MHz)	256/8	54/15	16/0
Due	AT91SAM3X8E (84 MHz)	512/96	54/12	12/2

2 Conexión de sensores e actuadores.

2.1 Tipoloxía de E/S.

As entradas/saídas do sistema poden ser analóxicas ou dixitais, para o que a linguaxe incorpora as seguintes funcións básicas:

a) E/S dixitais:

- _ pinMode (pin, modo); úsase para configurar un pin dixital como entrada (input) ou saída (output).
- _ digitalWrite (pin, valor); Escribe un valor no pin que pode ser (HIGH =5 V) ou (LOW =0 V).
- _ digitalRead(pin); devolve o estado lóxico dun pin dado.

b) Entradas analóxicas:

- _ analogRead(pin): Devolve un enteiro do convertedor analóxico dixital unipolar (10 bits valores de 0... 1023).
- _ analogReference(tipo): Permite cambiar a referencia de tensión xenérica (5 V) por unha referencia externa.

c) Saídas analóxicas:

- analogWrite(pin, valor): Genera una onda modulada en anchura de pulso PWM para así poder controlar actuadores analógicos. La resolución es de 8 bits (valores de 0...255).

Como pode verse, presenta un repertorio moi compacto para manexar gran cantidade de sensores e actuadores útiles nos sistemas automáticos e domóticos. Tamén se poderían usar sensores "intelixentes" que se comunicarían a través de portos serie, I2C, protocolo inarámico ZigBee, etc.

Nos seguintes apartados indicaranse unha serie de sensores e actuadores útiles nos sistemas a tratar.

2.2 Sensores.

Existen gran cantidade de sensores, pero analizarase brevemente unha selección de sensores económicos útiles nos sistemas domóticos.

- **Sensor de temperatura.**

O LM35 é un sensor de temperatura cunha precisión calibrada de 0.5°C . O seu rango de medición abarca dende -55°C ata 150°C . A saída está linealizada e cada grao centígrado equivale a 10mV. Polo tanto a lectura realízase simplemente mediante unha entrada analóxica.

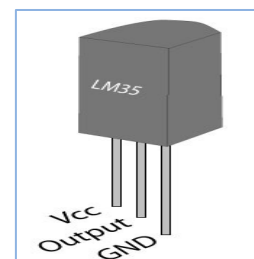


Figura 2-1 .-Sensor LM35

- **Sensor de luz.**

Aínda que existen distintos sensores de luz, considérase interesante o uso dunha fotorresistencia LDR integrada nun divisor de tensión, como pode verse na figura 3-2. Deste modo pódese calibrar para a súa lectura cunha entrada analóxica.

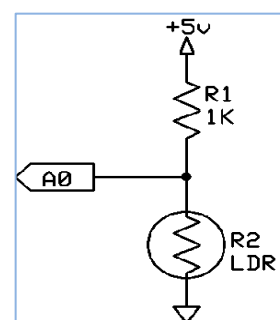


Figura 2-2 .-Montaxe LDR

- **Sensor de movemento PIR.**

Mediante un sensor pasivo de infravermellos (PIR) integrado como o da figura 3-3, pódese detectar a presenza de persoas ou animais nun cuarto. Este sensor pode lerse directamente mediante unha entrada dixital.



Figura 2-3 .- Sensor Pololu SE-10

- **Sensor de humidade.**

O sensor elixido é o da Fig. 3-4, o cal permite unha lectura de humidade relativa do 0 -100% cunha precisión do 2%. A transmisión de sinal realízase de forma dixital a través dun só fío, permitindo transmitir a 20 m de distancia.



Figura 2-4 .-Sensor Adafruit DHT-22

- **Sensores de gases.**

O fabricante Hanwei Electronics proporciona unha gama de sensores de gases para medir monóxido de carbono, propano, metano, alcohol, etc. moi versátiles. A lectura ha de realizarse mediante unha entrada analóxica, así como unha saída dixital para gobernar un sistema de prequentamento do sensor. Debido a este prequentamento, a cadencia máxima é dunha lectura por minuto.



Figura 2-5 .-Sensores de gases Hanwei

- **Sensor de corrente non invasivo.**

O sensor XiDi SCT-013 é capaz de medir ata 30 amperios sen contacto. Funciona como unha pinza amperimétrica, xerando unha tensión proporcional á corrente medida nun rango de 0-1 V. Pode ser útil para a optimización da demanda de enerxía eléctrica.



Figura 2-6 .- Sensor XiDi SCT-013

- **Sensor de presión barométrica.**

O sensor de presión barométrica BMP085 de alta precisión ten un rango de medida entre 300 e 1100 hPa cunha marxe de erro de tan só 0.03 hPa. Está deseñado para ser conectado directamente a un microcontrolador mediante a súa interface I2C.



Figura 2-7 .- Sensor BOSCH BMP085

2.3 Actuadores.

Dentro do campo da automática e a domótica existen infinidade de actuadores, pero moitos deles baséanse no control de motores (electroválvulas, persianas, trapelas, etc.) e no control da tensión de alimentación (iluminación, calefacción, etc.). Por iso, verase o control de elementos xenéricos que poderán formar parte de sistemas moi variados.

- **Relés e contactores.**

Son os elementos de control todo/nada máis usados tradicionalmente. Contrólanse mediante unha saída dixital debidamente amplificada por un transistor funcionando en conmutación e protexida cun díodo de libre circulación para evitar danos no transistor. Existen "shields" para inserir nas placas Arduino como a da Fig. 3-8, a cal inclúe 4 saídas a relé cos seus bornes de conexión sen necesidade de engadir máis elementos.

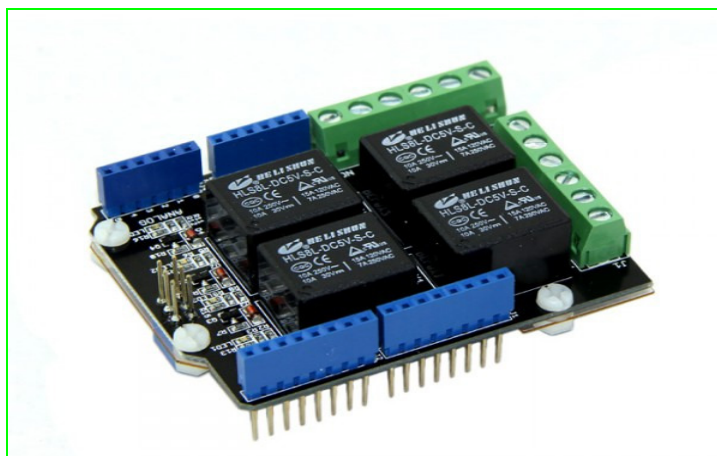


Figura 2-8 .- Placa Relé Shield

- **Control electrónico de motores de corrente continua.**

Ademais de poder controlarse con relés, é cada vez máis habitual o control con circuítos electrónicos que permiten a regulación de velocidade e sentido de xiro de forma moi sinxela. Un exemplo é o controlador L293, o cal permítenos controlar dous motores de forma independente a través de 3 patillas cada un:

- 2 saídas dixitais para controlar o sentido de xiro.
- 1 saída PWM para a consigna de velocidade.

Na Fig. 3-9 pode verse o diagrama de conexións para controlar un motor. Existen distintos controladores que soportan distintas correntes e tensións de control segundo as necesidades.

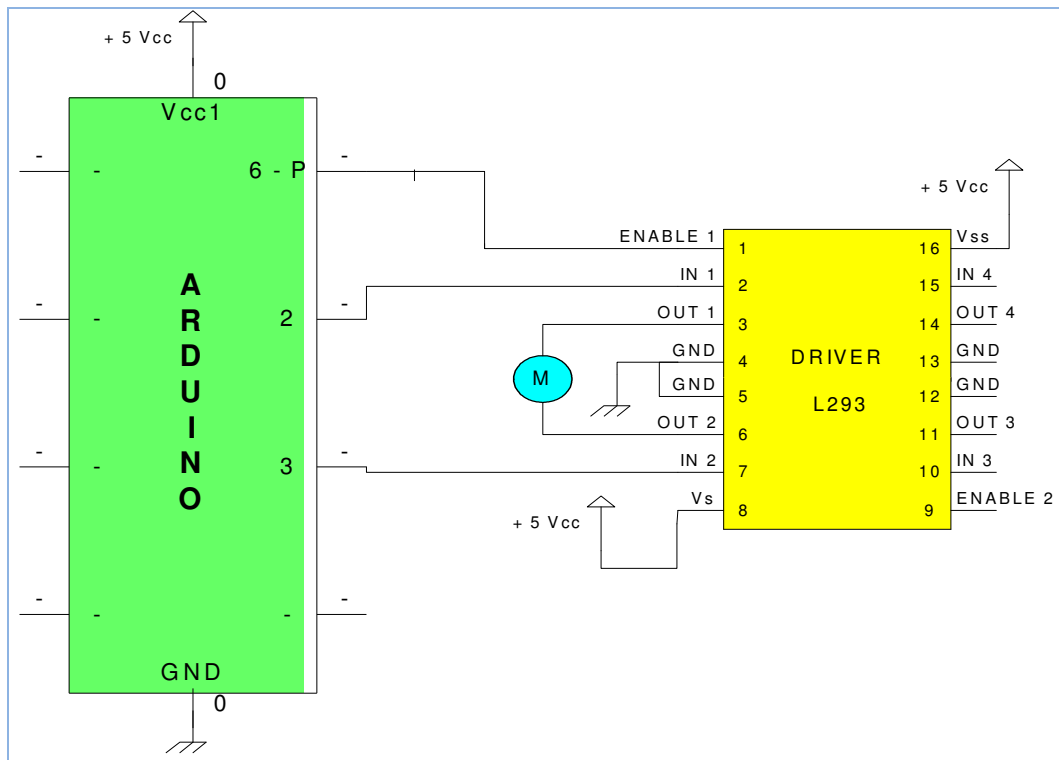


Figura 2-9 .- Control de motor C.C. con driver L293

- **Servomotores.**

Os servomotores poden controlarse a través dun único pin. Dado que os servomotores se controlan con sinais PWM con características específicas, o sistema Arduino conta cunha librería [Servo.h](#) que permite manexalos de forma moi simple. Na Fig. 3-10 pódese ver a forma de onda PWM que usan.

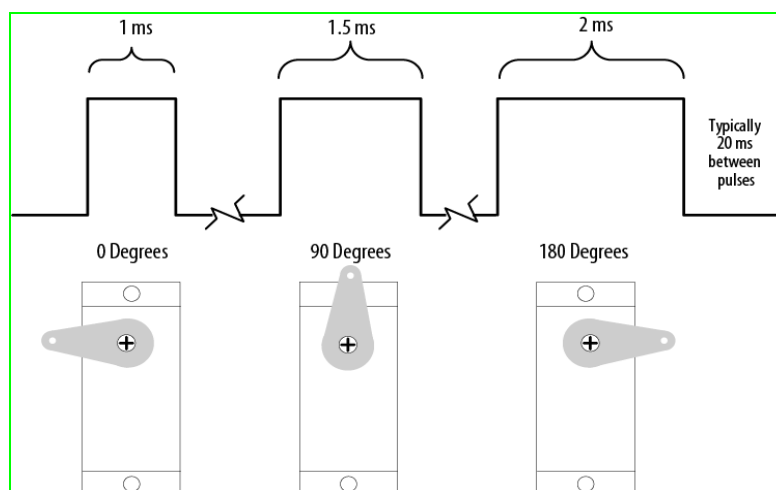


Figura 2-10 .-Control de un servomotor.

- **Motores Paso a Paso.**

O motor paso a paso é un dispositivo electromecánico que converte unha serie de impulsos eléctricos en desprazamentos angulares discretos, o que significa é que é capaz de avanzar unha serie de graos (paso) dependendo das súas entradas de control.

Deste modo permiten un control de posición moi preciso e fiable. Os servomotores poden ser unipolares ou bipolares. Na Fig. 3-11 pode verse un servomotor bipolar con conexión a 4 fíos.

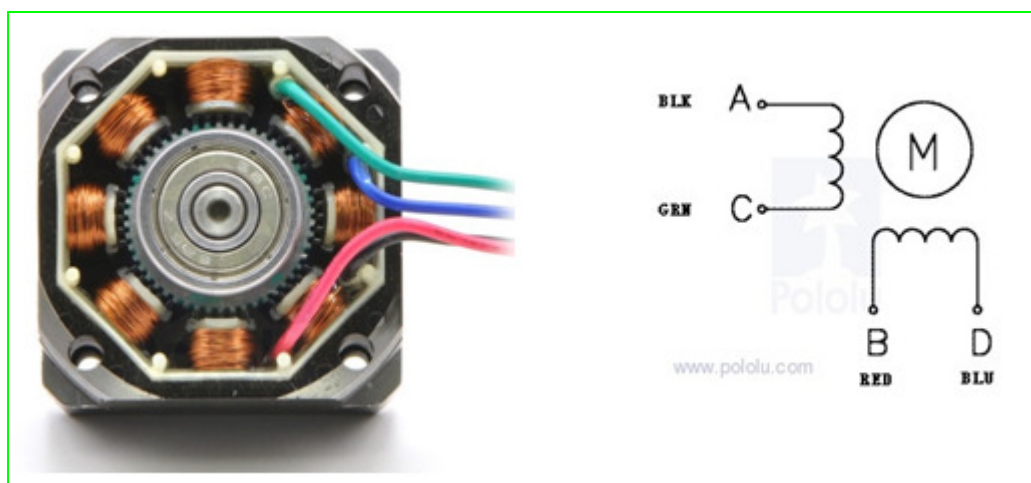


Figura 2-11 .- Motor Paso a paso de 4 fíos.

O control realízase aplicando impulsos aos debandados do motor seguindo unha secuencia dada, para o que o sistema Arduino conta cunha librería [Stepper.h](https://www.paulstoff.com/stepper/) que simplifica a xeración da devandita secuencia.

- **Control de iluminación.**

Tanto os balastros regulables de tubos fluorescentes coma as tiras de iluminación LED regulables en intensidade e cor, adoitan empregar como consigna un sinal analóxico de 0-10 V. Como se viu nas especificacións de Arduino a placa non dispón dunha saída axeitada, polo que será preciso utilizar adaptador axeitado. O fabricante Winkhel dispón do módulo WK-700 que pode verse na Fig. 3-12, o cal ten as seguintes prestacións:

- 4 saídas analóxicas 0-10 V.
- Recepción de consignas por bus I2C, dispoñible en Arduino.
- Montaxe en carril DIN.



Figura 2-12 .- Regulador Winkhel WK-700

2.4 Interfaces HMI.

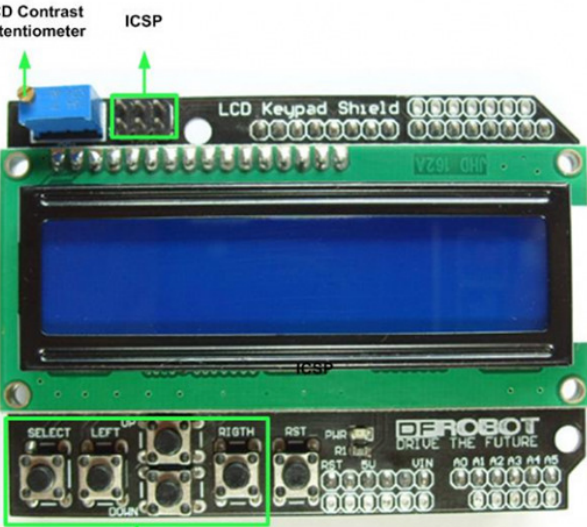
Os interfaces humano-máquina destes sistemas serán fundamentalmente os interruptores, potenciómetros de consigna, pantallas LCD. Os interruptores e interruptores poderán ser calquera modelo estándar, polo que aquí se tratarán só as pantallas e botoeiras que se poderán conectar directamente a Arduino.

- LCD Keypad Shield

Os displays estándar para Arduino utilizan un controlador Hitachi HD44780, o cal utiliza 6 saídas dixitais. O fabricante DF-Robot proporciona unha tarxeta que se insire sobre a placa de Arduino a cal proporciona:

- Display alfanumérico de 2 filas e 16 columnas.
- Botoeira con 5 interruptores asociados a un divisor de tensión, de modo que a lectura se realiza a través dunha única entrada analóxica.

Na Fig. 3-13 pode verse o seu aspecto, así como os pines que ocupa na placa de Arduino, os cales deixarán de estar dispoñibles para usar como L/S de sensores/actuadores.



Pin	Function
Analog 0	Button (select, up, right, down and left)
Digital 4	DB4
Digital 5	DB5
Digital 6	DB6
Digital 7	DB7
Digital 8	RS (Data or Signal Display Selection)
Digital 9	Enable
Digital 10	Backlit Control

Figura 2-13 .-LCD Shiled DF-Robot

Para manexar o display, dispense da librería LiquidCrystal.h .

- **Display de comunicación serie.**

Se trata de displays que se manejan a través de un único pin, mediante transmisión de comandos serie.

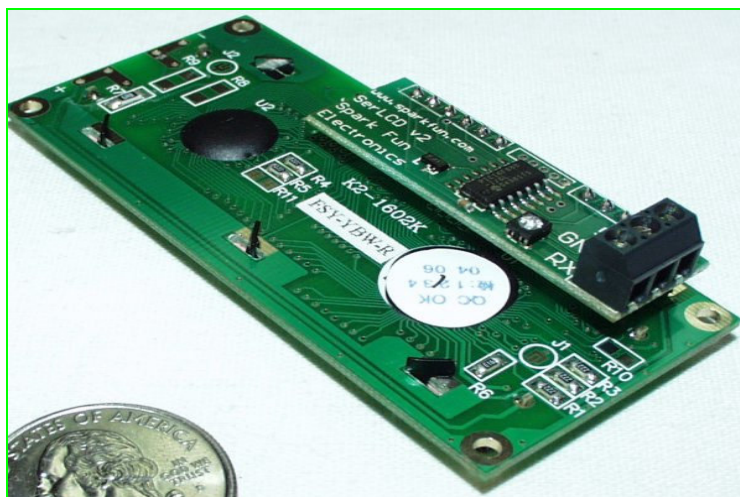


Figura 2-14 .- Display serial Sparkfun

Os devanditos displays dispoñen de tres pins:

- _ Rx: Recepción serie (entrada á pantalla). Niveis de tensión TTL, 9600 bps, 8 bits, 1 stop, sen paridade.
- _ GND: Masa común.
- _ Vcc: 5 V cun consumo de ata 60 mA.

É importante ter en conta que utiliza niveis TTL, polo que non é apta para conectar a canles serie con niveis de tensión RS-232.

A placa incorpora un microcontrolador que traduce os comandos serie e manexa o display. Os comandos serie enviaranse en formato ASCII.

A placa de Arduino Diecimila incorpora unha soa UART, polo que só dispón dun porto serie. Non obstante, mediante a librería [SoftwareSerial.h](https://www.sparkfun.com/products/1002) pódese crear unha UART virtual en calquera pin dixital, polo que poderemos manexar varias pantallas simultaneamente.

3 Comunicacóns.

As comunicacóns son fundamentais en calquera sistema domótico. Poderán permitir tanto a monitorización remota do sistema coma a comunicacón de campo con sensores e actuadores intelixentes.

3.1 Comunicaciones serie asíncronas.

A nivel industrial, as comunicacóns serie asíncronas máis usuais son RS-232 e RS-485. Presentan as seguintes características:

- **Canal RS-232.**

Trátase da canle implementar de forma nativa nas placas de Arduino orixinais. Só permite a conexión punto a punto a distancias máximas de 30 m con velocidades ata 20 kbps.

As placa de Arduino Diecimila dispón dun porto serie nativo utilizado para a comunicacón co porto USB, mentres que a versión Mega dispón catro portos. Para suplir esta limitación, o IDE de Arduino proporciona a librería [SoftwareSerial.h](https://www.arduino.cc/en/reference/software/SoftwareSerial) coa que se pode crear unha UART por software en calquera par de pines dixitais. Deste modo, podemos ampliar os portos serie asíncronos dispoñibles de forma nativa no microcontrolador. É importante destacar que se se utilizan varios portos serie software, só un pode recibir datos a este simultaneamente.

- **Canal RS-485.**

Trátase dunha versión con transmisión balanceada, o que permite percorrer distancias de 1.2 Km a unha velocidade de transmisión de 100Kbps. Outra vantaxe clave é a de poder compartir o bus ata 128 dispositivos, mediante unha topoloxía mestre-esclavos.

As placas de Arduino non inclúen o porto RS-485 de forma nativa, pero o fabricante Winkhel inclúeo nun sistema baseado no Arduino Diecimila á que lle engadiron uns periféricos, fonte de alimentación e carcasa para carril DIN. Pode verse na Fig. 4-1.



Figura 3-1 .- Placa Winkhel WK-100

Mediante o bus RS-485 poderemos comunicarnos con multitude de dispositivos que empregan o coñecido protocolo MODBUS. O IDE oficial de Arduino polo momento non inclúe librerías oficiais para este protocolo, pero na comunidade de usuarios existe moita información.

As placas de Arduino non inclúen o porto RS-485 de de forma nativa, pero o fabricante Winkhel inclúeo nun sistema baseado no Arduino Diecimila á que lle engadiron uns periféricos, fonte de alimentación e carcasa para carril DIN. Pode verse na Fig. 4-1.

3.2 Comunicaciones Ethernet y Wi-Fi.

As comunicacións Ethernet convértense en esenciais no fogar dixital actual, polo que resulta moi interesante integrar o sistema domótico coa rede de datos existente. Para iso, existen "shields" para inserir nas placas de Arduino e mesmo placas completas con Ethernet integrado.

Como veremos posteriormente, isto facilitará a monitorización e o telecontrol dos sistemas automatizados.



Figura 3-2 .-Módulo Ethernet en Arduino

O chip que goberna a comunicación é un Wiznet W5100, permitindo a comunicación en redes 10/100 Mbps. Admite a comunicación TCP e UDP, cun máximo de 4 conexións socket simultáneas.

O conector é RJ-45 e soporta PoE (transmisión de enerxía pola rede Ethernet) e conta con varios LEDs de estado que proporcionan información interesante do estado do enlace de datos.

As tarxetas Ethernet comunícanse coa placa de Arduino a través do protocolo SPI, ocupando os pines indicados na Fig. 4-3 segundo sexa a placa de Arduino.

Tabla 3-1 .-Pins usados na tarxeta Ethernet

Señal bus SPI	Placa Arduino estándar	Arduino Mega
SCLK (clock)	13	52
MISO (saída de datos)	12	50
MOSI (entrada de datos)	11	51
SS (selección de esclavo)	10	53

Proporciónase a librería [Ethernet.h](#) para facilitar o desenvolvemento de servidores e clientes Ethernet.

Ademais estas tarxetas inclúen unha rañura para unha tarxeta de memoria SD, na que poderemos ler/escribir mediante a librería [SD.h](#). De igual modo existen tarxetas Wi-Fi que permiten a conexión a redes inalámbricas IEEE 802.11. Existen modelos con antenas integradas ou con conectores para antenas de maior alcance. Para manexalas dispoñemos da librería [WiFi.h](#).



Figura 3-3 .-Wi-Fi shield DF-Robot

3.3 Bus I2C.

Philips deseñou o bus I2C no ano 1980 para a comunicación entre circuitos integrados, pero dada a súa versatilidade foise estendendo a multitude de dispositivos: sensores intelixentes, memorias, perifera distribuída, etc. A velocidade do bus orixinal era de 100 Kbps, pero na versión actual pode alcanzar 3.4 Mbps.

A través dunha liña de datos (SDA) e outra de reloxo (SCL) permite a comunicación entre un dispositivo mestre e varios escravos, como pode verse na Fig. 4-5. Permítese mesmo a conexión de varios dispositivos másters, xa que incorpora un mecanismo de detección de colisións.

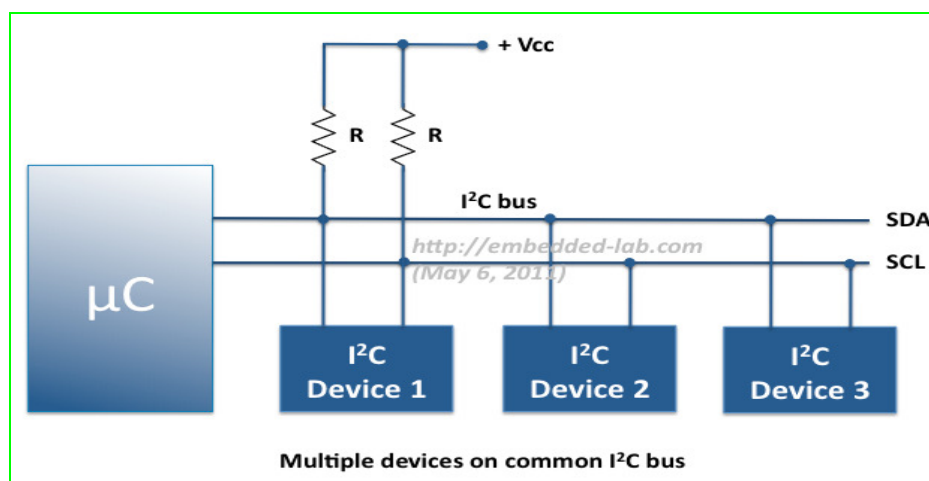


Figura 3-4 .- Esquema de conexión de varios dispositivos I2C

Cada dispositivo conta cunha dirección de 7 bits configurable por software.

Tabla 3-2 .- Pines I2C en las placas Arduino

Modelo Arduino	Pines I2C
Uno, Ethernet	A4 (SDA), A5 (SCL)
Mega2560	20 (SDA), 21 (SCL)
Leonardo	2 (SDA), 3 (SCL)
Due	20 (SDA), 21 (SCL), SDA1, SCL1

Cóntase coa librería [Wire.h](http://www.arduino.cc/en/reference/wire) para manexar a comunicación.

Hai moitos dispositivos que usan este protocolo: sensores intelixentes variados, controladores de balastros electrónicos, módulos de L/S remotos, etc.

3.4 ZigBee.

Zigbee é un protocolo de comunicacións inarámico baseado no estándar IEEE 802.15.4 creado por unha alianza de empresas "Zigbee Alliance". Este estándar permite realizar comunicacións inarámicas en dispositivos de baixo consumo.

É moi útil en redes de sensores industriais e domóticos. Os módulos Xbee do fabricante MaxStream integran nun único dispositivo un trasmisor-receptor Zigbee e un procesador, tal e como pode verse na Fig.4-5. A antena pode ir integrada no módulo ou conectarse unha antena externa.



Figura 3-5 .- Aspecto e patillaxe do módulo Xbee

As principais características dos devanditos módulos son:

- Utilizan a banda 2.4 GHz, dispoñendo de 16 canles configurable.
- Velocidade de transmisión: 256 kbps.
- Alcance en campo aberto: 100 metros para os módulos Xbee e 1.6 Km para Xbee-PRO.
- 9 pines de entradas/saídas analóxicas e dixitais.
- Canle serie incorporada.
- Baixo consumo: 50 mA cando transmiten e menos de 10 uA cando están en modo "sleep".

Actualmente existen dúas versións dos devanditos módulos, os cales son incompatibles entre se:

- Serie 1: Pensado para formar redes punto-punto e punto-multipunto.

- Serie 2: Diseñados para aplicacións que requiren repetidores ou unha rede mallada (mesh).

Os nodos Xbee poden configurarse para realizar tres funcións: Coordinador, router e dispositivos finais. Deste modo poden crearse distintas topoloxías de rede da Fig. 4-6.

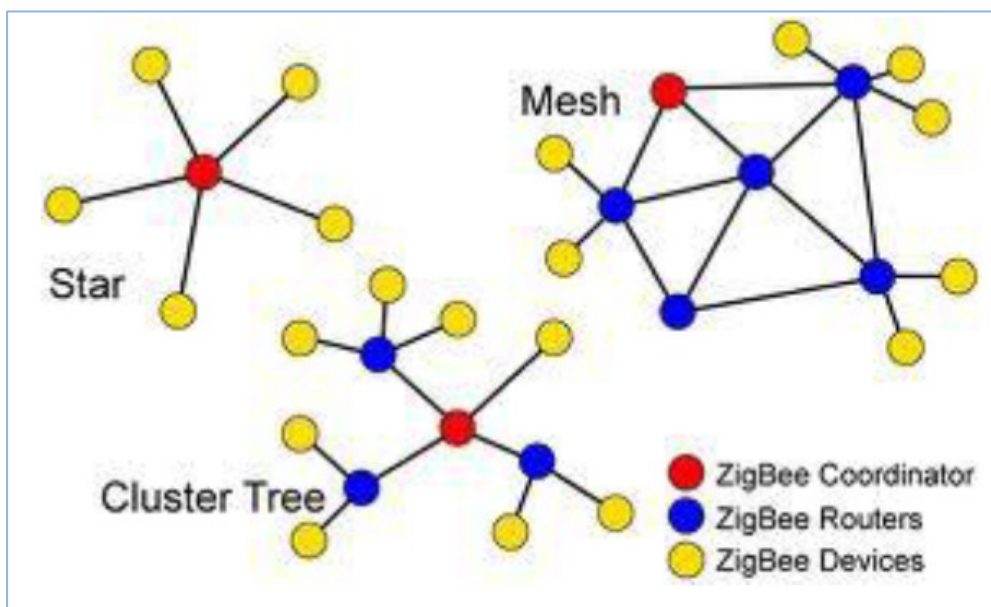


Figura 3-6 .-Topoloxías de red Xbee.

Non require ningunha programación específica no código de Arduino, pero é preciso configurar os módulos segundo as necesidades do proxecto. Na páxina de [DigiXbee](#) encóntranse guías de uso e exemplos.

Deste modo, poderemos situar sensores e actuadores remotos comunicados coas placa de Arduino que governe cada zona.

3.5 Comunicaci3ns GSM

Cando se necesite realizar a xesti3n remota mediante a rede telef3nica celular, pode utilizarse unha placa con conexi3n " GSM Shield" como a da Fig.4.7.



Figura 3-7 .- Tarjeta GSM Shield

Esta tarxeta 3 capaz de enviar e recibir SMS, realizar chamadas telef3nicas e conexi3ns de datos a trav3s de GPRS. Para iso 3 necesario inserir unha tarxeta telef3nica SIM de calquera operador con cobertura na zona.

Esta tarxeta disp3n dun modem que transfere datos serie as3ncronos a trav3s dos pines 2 e 3 mediante unha UART virtual, polo que quedan dispo3nibles os pines serie nativos da placa dispo3nibles para outros usos. Para facilitar o uso, nas versi3ns do IDE de Arduino 1.04 e superiores incl3use a librer3a [GSM.h](http://www.arduino.cc/en/reference/gsm), a cal incl3e comandos que nos permiten abstraernos dos comandos entre o modem e a tarxeta SIM.

4 Integración en sistemas domóticos.

4.1 Arquitecturas de comunicación.

Unha vez vistos os principais elementos de control, sensores, actuadores, interfaces co usuario e comunicacións xa podemos integrar todos os elementos necesarios e engadirlle a posibilidade da monitorización e a xestión mediante dispositivos móbiles.

Decidiuse utilizar unha arquitectura distribuída, dispoñendo en cada cuarto dunha placa de Arduino que se encargará de gobernar esta. As placas de todas as estanzas da vivenda estarán unidas a través da rede de datos Ethernet ou Wi-Fi, como pode verse na Fig. 5-1.

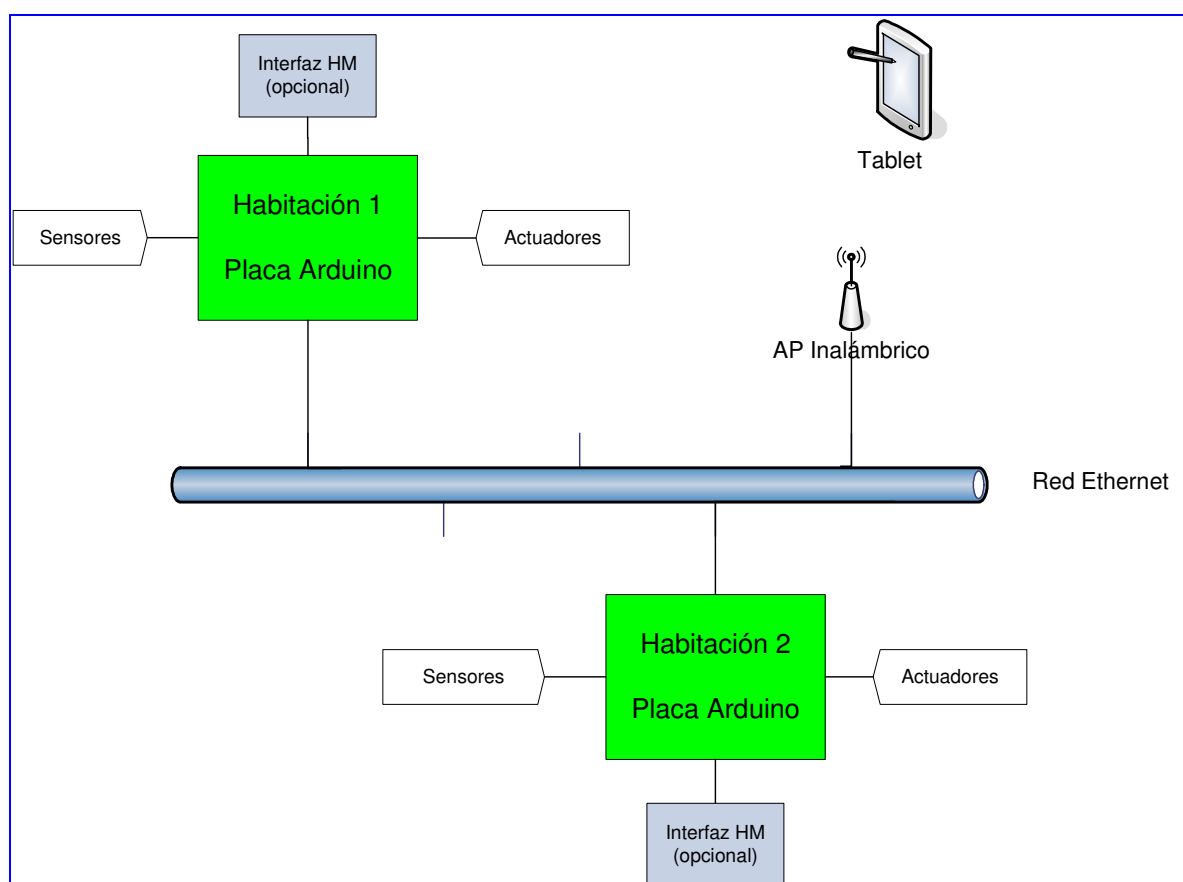


Figura 4-1.-Arquitectura de comunicación domótica

Dentro de cada estancia poderanse instalar os seguintes elementos:

- Fonte de alimentación.
- Placa de Arduino con conexión Ethernet ou Wi-Fi.
- Sensores analóxicos ou intelixentes I2C.
- Interruptores para o control manual.
- Tarxetas de saídas a relé e/ou saídas analóxicas.
- Outros elementos de control: electroválvulas, variadores de velocidade, etc.
- Módulos Xbee para a comunicación de elementos de campo remotos.

A rede de datos doméstica non se utilizará como bus de campo, senón para a monitorización e tele xestión das estancias. O control en tempo real executarase en cada nodo, xa que a rede de datos non garante un tempo de resposta determinado.

Este esquema ten as seguintes vantaxes:

- _ Aproveitase a rede de datos existente no fogar dixital como canle de comunicación entre os nodos.
- _ Poderase utilizar a rede inarámica Wi-Fi naqueles lugares onde non teñamos puntos de conexión á rede.
- _ Non se depende dun elemento de control central, polo que aínda que haxa unha avaría na rede de datos ou noutros nodos, as placas de cada estancia poderán seguir operando de forma autónoma.

En canto á monitorización e telexestión do sistema, pode codificarse en cada placa de Arduino un servidor Web de forma que poidamos ler remotamente información dos sensores e/ou actuar sobre as saídas.

A solución de que cada nodo funcione como un servidor Web resulta excesivamente pesada para unha placa microcontroladoras con prestacións relativamente reducidas, ademais de xerar un tráfico de rede excesivo. Polo tanto, investigáronse alternativas que permitan a xestión de forma lixeira e que inclúan interfaces gráficas para o usuario.

4.2 Software de xestión: Souliss

A opción que se considerou máis interesante para xestionar o sistema domótico é [Souliss Automation](#), o cal é tamén software libre. Trátase dun xogo de librerías ou "framework" que permite que distintos nodos compartan información de forma lixeira, rápida e fiable.

Un nodo é un dispositivo xenérico capaz de executar o "framework", como pode ser un microcontrolador AVR (Arduino ou outros) ou outros dispositivos con sistema operativo Android (teléfonos móbiles, media centers, tablets, etc.) Como pode observarse na Fig. 5-2, axústase moi ben á arquitectura de funcionamento elixida e dispón de aplicacións gráficas para instalar en "tablets ou smartphones". Na devandita figura obsérvase un sistema simple de control dun punto de iluminación de forma local (interruptor) ou remota (dispositivo móbil) mediante unha placa Arduino con Ethernet.

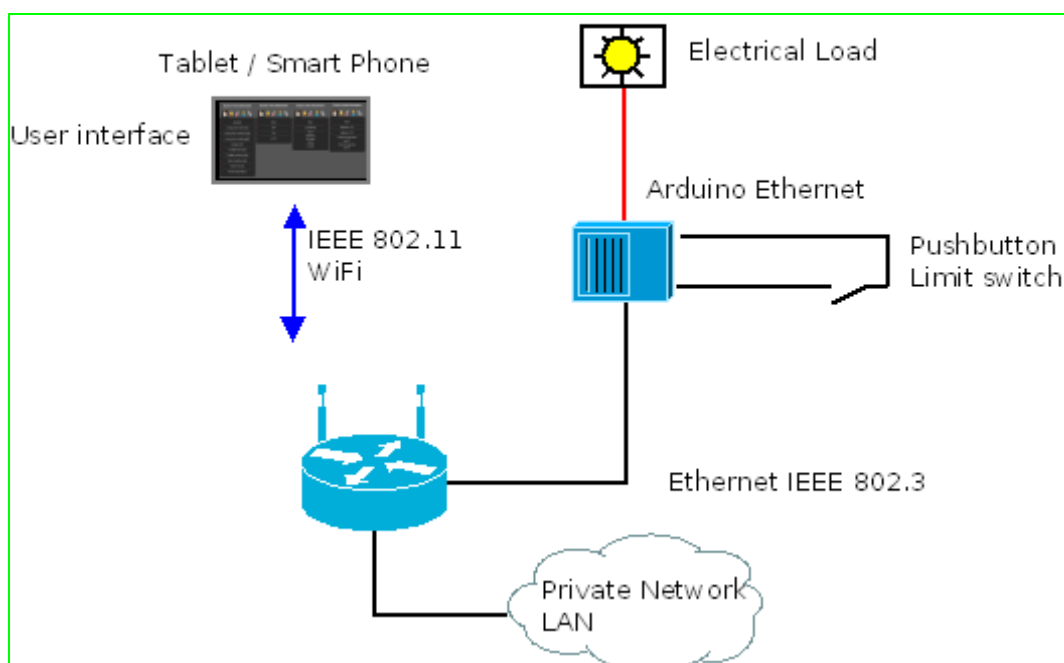


Figura 4-2 .-Arquitectura Souliss simple

A continuación estudaranse as partes imprescindibles para a posta en marcha:

- **Estrutura de datos.**

Souliss emprega un protocolo de comunicación chamado MaCaCo, o cal é binario e está baseado en eventos. Permite a comunicación punto a punto entre os nodos, podendo actuar como nodo un dispositivo con sistema operativo Android, permitindo unha interacción rápida e de baixo consumo de batería grazas ao mecanismo baseado en eventos.

Para iso, créase unha área de memoria compartida en cada nodo como a que pode verse na Fig. 5-3.

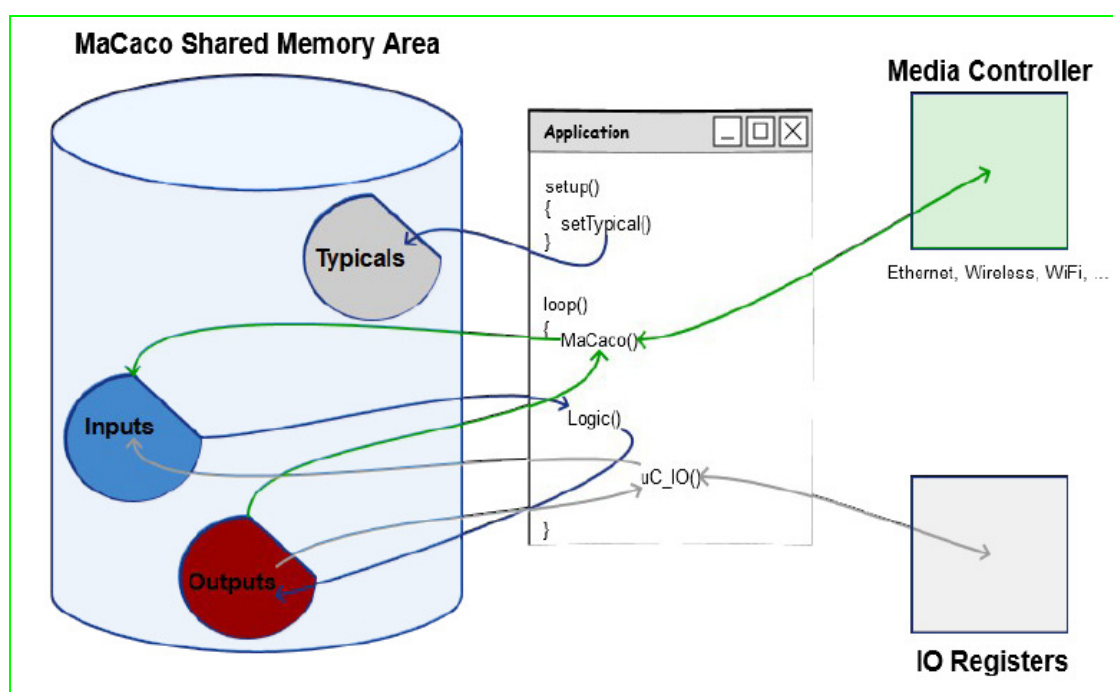


Figura 4-3 .-Memoria compartida na plataforma Souliss

Cada unha das áreas ten a seguinte función:

- **Typicals**: Indican unha funcionalidade lóxica estándar, a cal é indicada cun único número identificador. Exemplos: 12 (Relé con 1 entrada e 1 saída), 13 (relé con sensor PIR), 22 (porta motorizada), 31 (Control de temperatura), etc.
- **Inputs**: Área para manexar a información recibida doutros nodos ou da aplicación do propio nodo.
- **Outputs**: Área para manexar a información dirixida a outros nodos ou á aplicación do propio nodo.

- **Instalación de las librerías Souliss no entorno de Arduino.**

Para poder codificar programas en Arduino é preciso descargar o último [framework souliss](#) dispoñible e copialo dentro do directorio "arduinolibraries".

Unha vez que reiniciemos o ámbito, ademais das librerías aparecen unha lista de exemplos en "Arquivo -> Ejemplo-> Souliss"

A continuación será preciso realizar unha configuración do ámbito, a cal poderá entenderse mellor cun exemplo comentado.

4.3 Aplicación Souliss para o sistema operativo Android.

A aplicación Souliss é compatible con Android 2.3 ou superior e pódese descargar gratuitamente dende [Google Play Store](#), ocupando tan só 3 MB de memoria.

Mediante esta aplicación é posible monitorizar e xestionar todos os dispositivos de forma amigable, xa que dispón dun interface gráfico que simula o obxecto "Typical" ao que está asociado. Na Fig. 5-4 pode verse o aspecto do interface para un control temporizado de iluminación, un climatizador e unha tira de LEDs RGB.

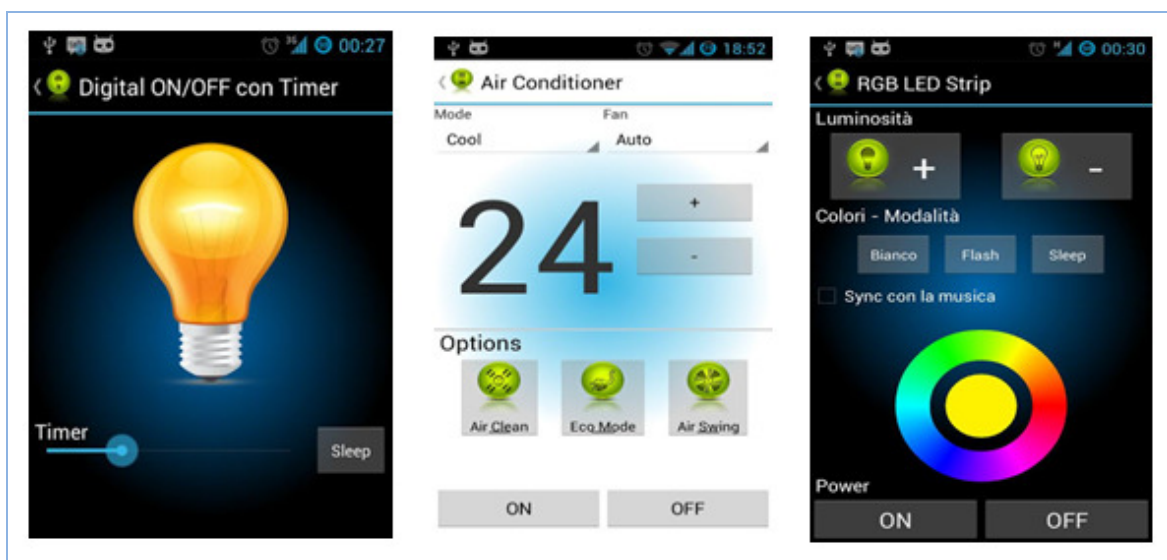


Figura 4-4 . Aspecto del interfaz de Souliss

Veremos a continuación os principais pasos a seguir para introducir o manexo da aplicación.

- **Instalación e configuración:**

A asignación de dispositivos "Typical" encontrarase configurada en cada nodo, polo que a configuración no dispositivo Android se reduce a dous pasos:

- _ Configurar a dirección IP de todos os nodos aos que queremos acceder. A aplicación comprobará a conectividade con cada nodo.
- _ Engadir os dispositivos existentes en cada nodo. Pulsando no botón "Souliss DB" encherá automaticamente unha base de datos local que contén toda a estrutura de datos da nosa rede domótica.

Dese modo, poderemos acceder a unha ventá como a da Fig.5-4 na que poderemos etiquetar os nodos: baño, cociña, salón, etc. Dende a pantalla de detalles de nodo é posible cambiar o nome de nodo e elixir unha icona de nodo personalizado.

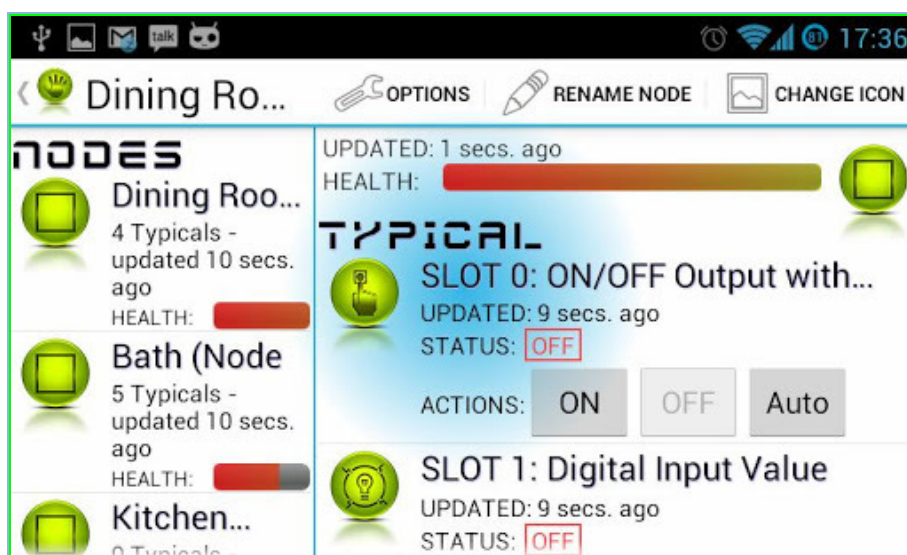


Figura 4-5 .- Ventá de configuración de Souliss

Dentro de cada nodo aparecerannos os dispositivos "Typical" existentes, así como o seu estado e a última vez que cambiou este. Dado que utiliza un protocolo baseado en eventos, a actualización realízase cando se realiza un cambio de forma local ou remota. Deste modo, consúmese pouca batería e tráfico de rede.

- **Manexo do entorno:**

Dentro da aplicación encontraremos 3 grupos de accións:

- _ Manual: Para cada dispositivo, móstranse os seus controis e / ou unha pantalla con máis detalle, por exemplo un gráfico (para o sensor analóxico). Pode verse o aspecto na Fig. 5-4
- _ Escenas: Conxunto de comandos que se executan en secuencia. Os comandos poden ser ou ben "masivos" ou "singre": os primeiros execútanse en todos os dispositivos do mesmo tipo, mentres que os comandos individuais só se aplican a un dispositivo determinado.
- _ Programas: Son secuencias de comandos que se poden executar segundo unha programación horaria, segundo a presenza do usuario na vivenda e condicionado ao estado dalgún sensor.

5 Bibliografía.

- <http://arduino.cc/>: Página oficial de Arduino.
- Arduino Cookbook. Autor: Michael Margolis. Editorial O'Reilly.
- [Monografías: Aplicaciones de Arduino](#) Autor: José Manuel Ruíz Gutiérrez.
- <http://www.souliss.net/> :Página oficial Souliss Automation.