

Disseny e implantació d'un servidor Web que permeti la configuració de paràmetres en el processador host per al mòdul Wi-Fi RTX4100

Carlos Saleta Pereda

Presentació del Treball Final de Grau de Technologies de la Telecomunicació





✦ www.smartcitizen.me

Index:

- Objectius de projecte
- Anàlisi Hardware Smart Citizen Kit 1.1 / 1.5
- Coneixement teòric
 - Protothreads
 - Comunicació SPI
 - Programació HTML
- Entorn de desenvolupament
 - IDE Arduino
 - Amelie SDK
- Desenvolupament
 - SOFTAP
 - SPI
 - WebServer
- Resultats
- Futurs desenvolupaments
- Conclusions

Objectius del projecte

- Expansió de funcionalitats del Firmware de Smart citizen per comunicar-se amb el mòdul Wi-Fi RTX4100.
- Creació Servidor Web
- Creació SoftAP
- Consulta de temperatures
- Enviament de comandes



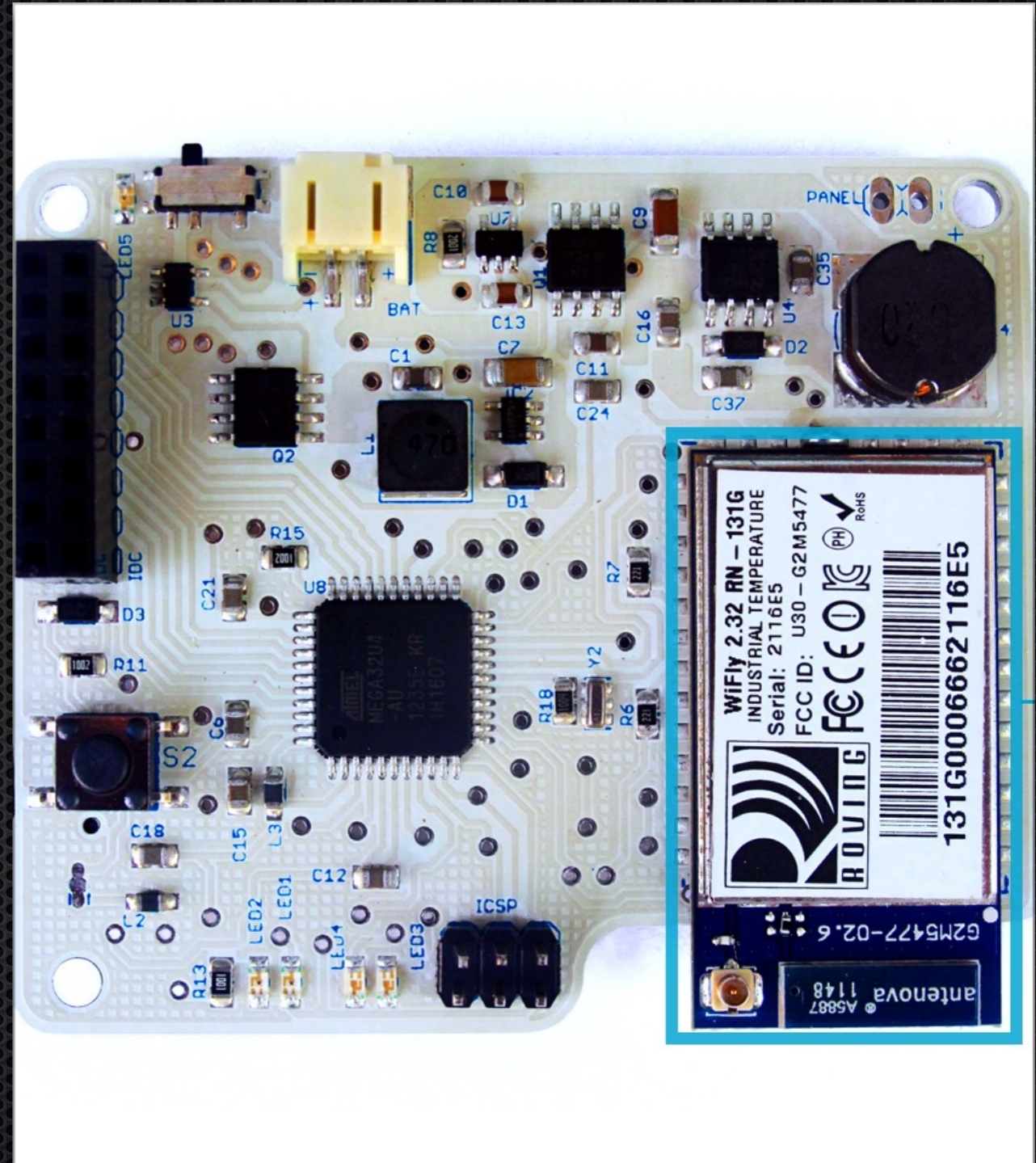
▪ Anàlisis Hardware Smart Citizen Kit 1.1 / 1.5

▪ Hardware SmartCitizen 1.1

- * ATmega32U4
- * Sensor I/O
- * Switch On/off
- * LiPo battery socket
- * Mòdul Wi-Fi RN-131

Característiques de funcionament RN131:

- * Temperatures de 0 -70°
- * Comandes limitades
- * Edició no disponible
- * Consum 210mA

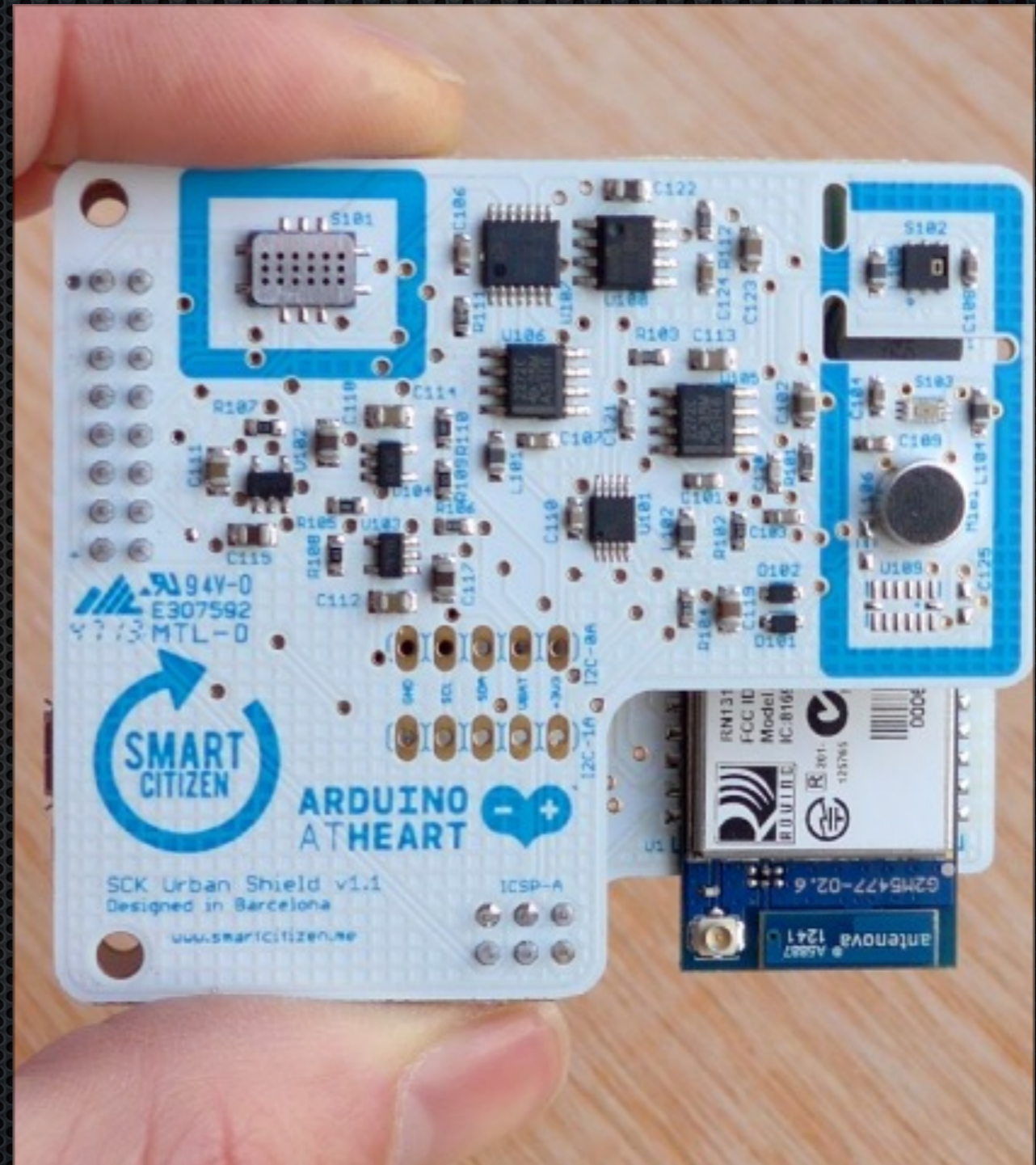


Smart Citizen Kit 1.1

✦ Anàlisi Hardware Smart Citizen Kit 1.1 / 1.5

✦ Hardware SmartCitizen 1.1

- * Sensor de Gasos
- * Sensor Temperatura i humitat
- * Sensor de luminància
- * Micròfon



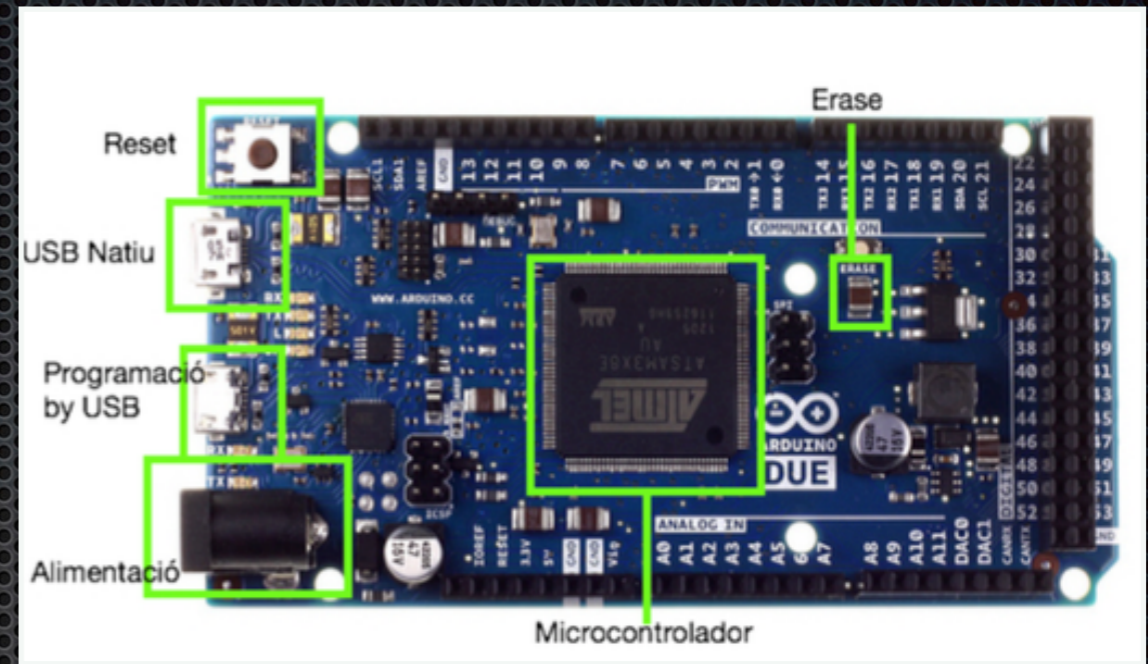
Smart Citizen Kit 1.1 Ambient Board

✦ Anàlisi Hardware Smart Citizen Kit 1.1 / 1.5

✦ Hardware SmartCitizen 1.5

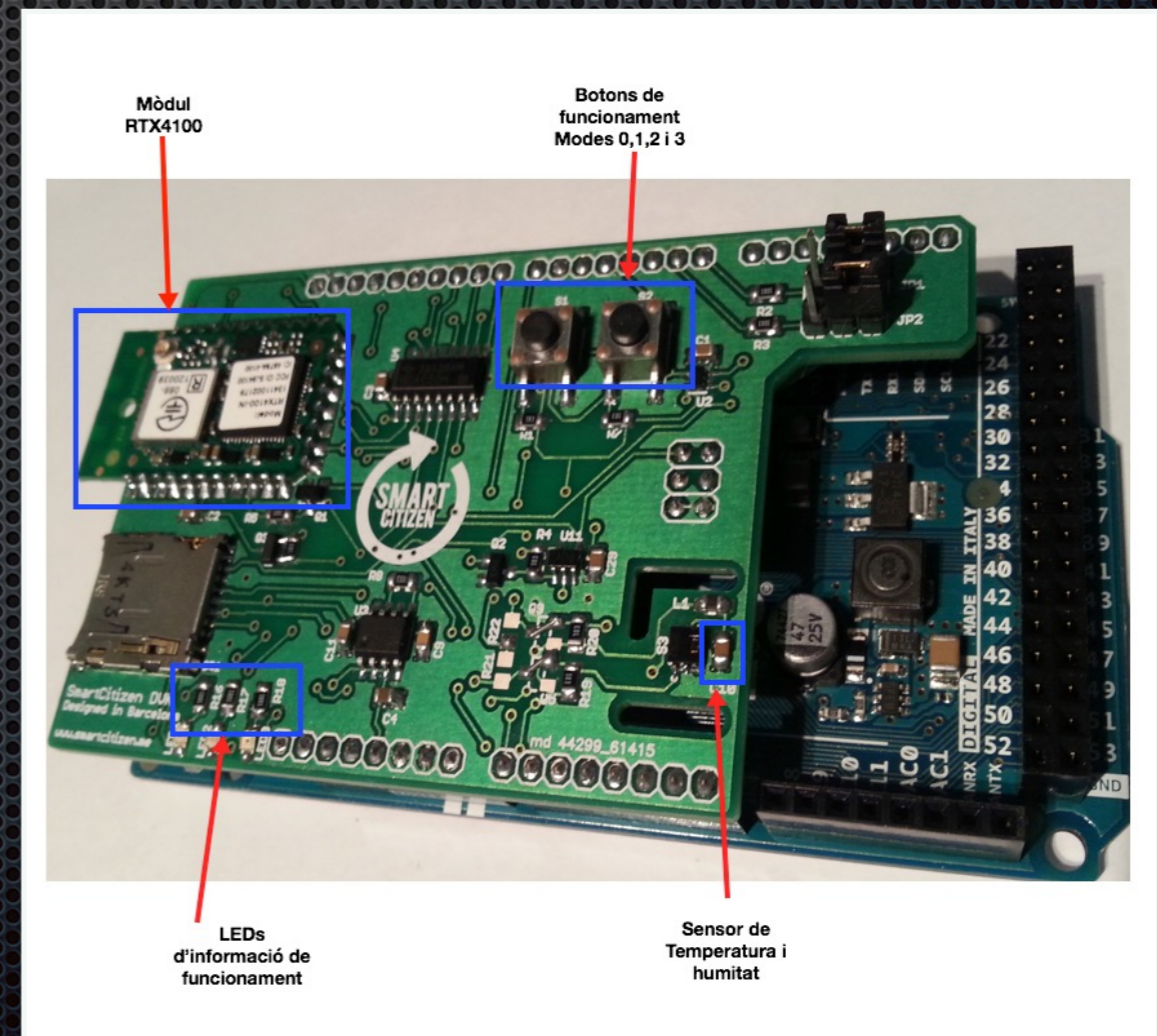
Part Arduino DUE

- *Reset
- *USB Natiu
- *Programació per USB
- *Alimentació
- *Microcontrolador
- *Polsador Erase



Part SCDVK

- *Mòdul Wi-Fi RTX4100
- *Polsadors S1 i S2 per seleccionar el mode de funcionament(4 modes)
- *Sensor de temperatura i humitat
- *LEDs d'informació de MODE.



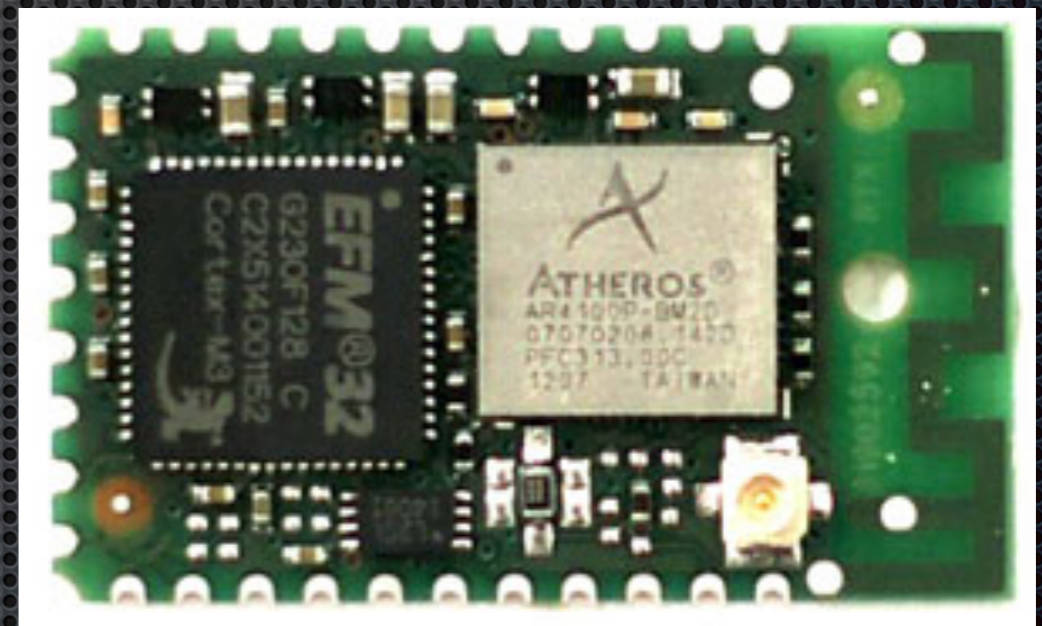
Smart Citizen Kit 1.5 , Adalt Arduino Due, sota SCDVK

✦ Anàlisi Hardware Smart Citizen Kit 1.1 / 1.5

✦ Hardware SmartCitizen 1.5

Part RTX4100

- * Microcontrolador EFM32G230
- * Chip Wi-Fi Atheros AR4100
- * Temperatures -45° a 85°
- * Permet la manipulació del Firmware mitjançant AmelieSDK



Mòdul Wi-Fi RTX 4100

❖ Coneixement teòric

❖ Protothreads

Conjunt de macros basat en accions, facilita la legibilitat i és més econòmic de mantenir, alguns exemples són PT_BEGIN, PT_WAIT_UNTIL, PT_YIELD_UNTIL, etc.

❖ Comunicació SPI

Protocol síncron, full-duplex, i permet la comunicació entre dispositius. Un dels dispositiu pren la tasca de “Master” i l’altre de “Slave.

Linies logiques:

- MOSI: Master Out Slave Input
- MISO: Master Input Slave Output
- CLK: Relotge de sincronia
- Slave Select: Selecciona i habilita un esclau

❖ Programació HTML5 i Protocol HTTP

HTML i CSS

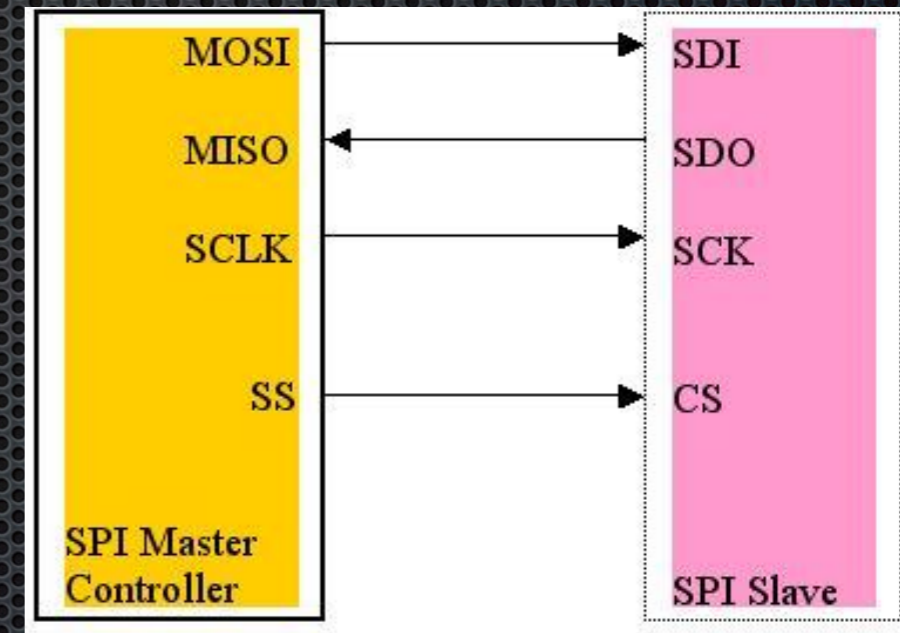
- Conèixer les classes, etiquetes “div” i “id”
- El funcionament de les “tab” i caixes
- El format del text
- Incrustació de fulla d’estils

Protocol HTTP

- Permet l’intercanvi d’informació mitjançant un servidor i un client amb els mètodes GET i POST

```
int a_protothread(struct pt *pt) {
    PT_BEGIN(pt);
    /* */
    PT_WAIT_UNTIL(pt, condition1);
    /* */
    if(something) {
        /* */
        PT_WAIT_UNTIL(pt, condition2);
        /* */
    }
    PT_END(pt);
}
```

Exemple Protothread



Esquema comunicació SPI



Esquema comunicació SPI

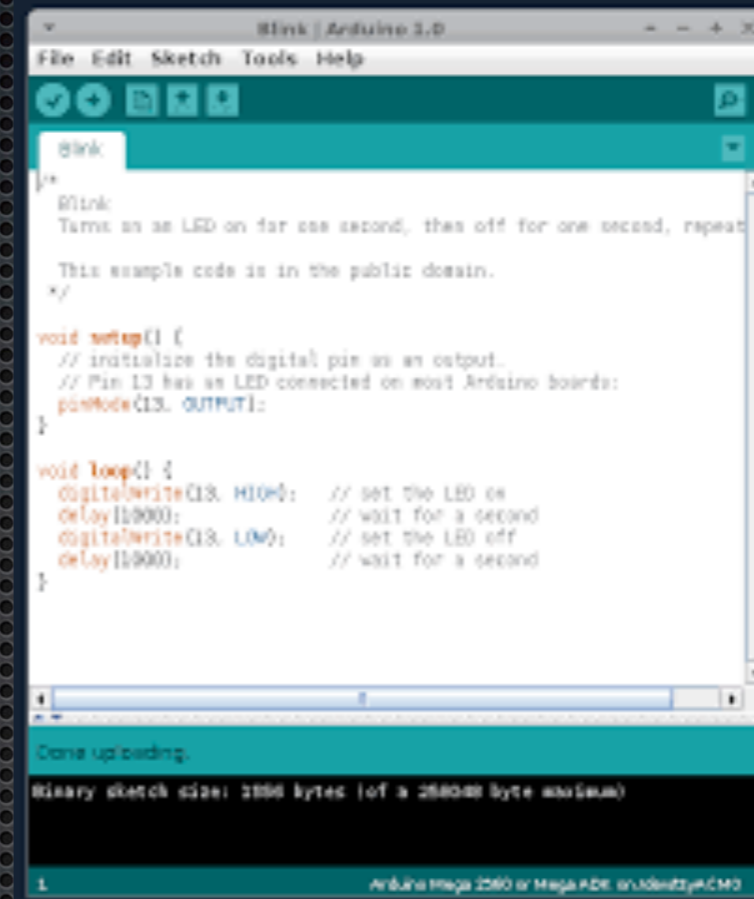
✦ Entorn de desenvolupament

✦ Arduino IDE

Una eina optimitzada per la creació dels “Sketch” d’un projecte per implementar funcionalitats dins de la placa Arduino, en aquest cas, el DUE.

El Firmware de Smart Citizen Kit es:

- SCDVK.h: Header de crides
- SCDVK.cpp: funcions implementades
- SmartCitizen.ino: inclou el setup i loop de creació del sketch per Arduino i que executa les funcions del arxiu .cpp
- Constants.h: Constants per donar facilitat en la programació



```
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

Arduino IDE



Arxius Firmware Smart Citizen Kit 1.5


Entorn de desenvolupament

Amelie SDK

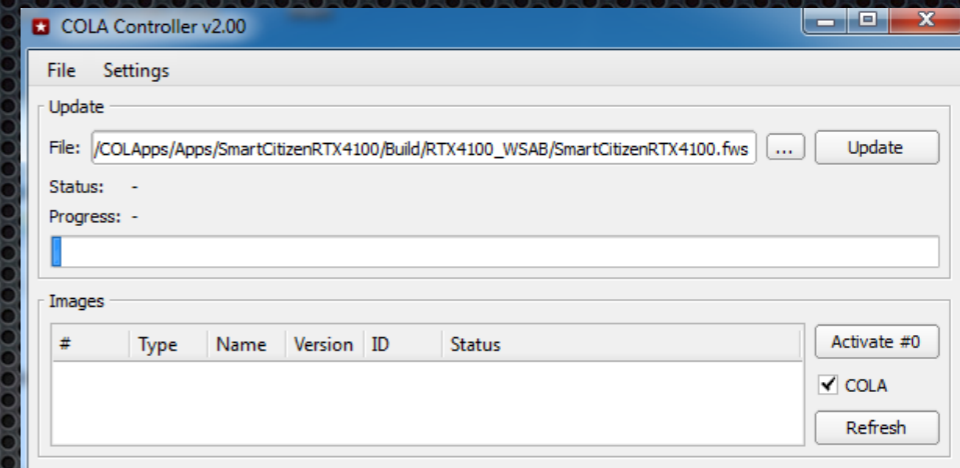
Conjunt d'aplicacions per a desenvolupar funcionalitats, anomenades COLApps, dins del mòdul Wi-Fi RTX4100.

Les eines que s'inclouen són:

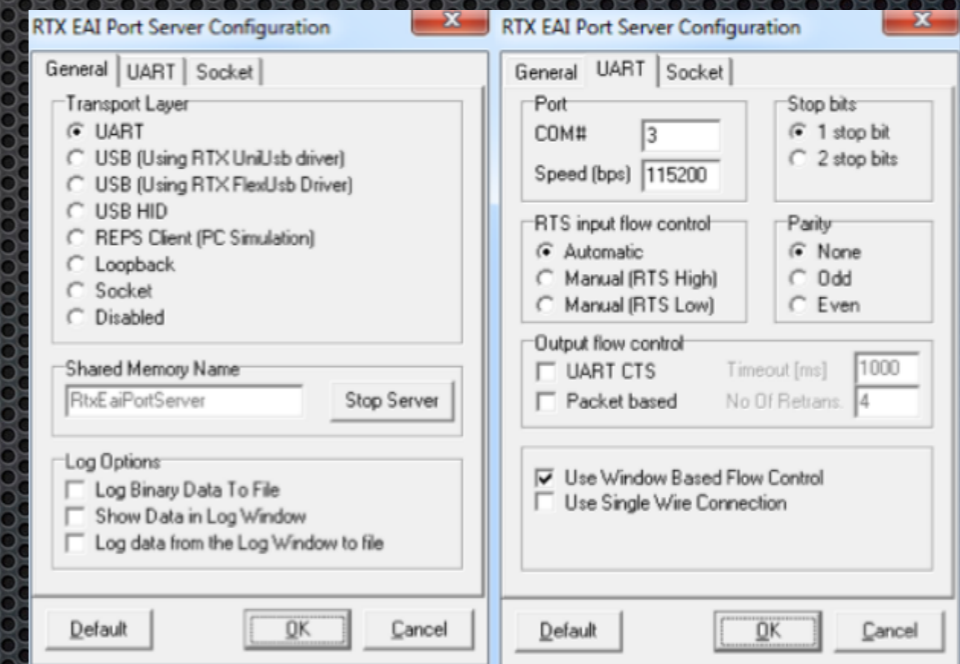
- Amelie COLAController: Ajuda a pujar les COLApps programades al mòdul Wi-Fi.
- Mail tracer: debug per veure el funcionament
- RTX EAI Port Server: Establiment de connexió amb el mòdul perquè seguidament es puguin pujar les aplicacions amb el COLAController

 gcc-arm-none-eabi-4_7-2013q3-20130916-win32.exe

gcc-compilador -> <https://launchpad.net/gcc-arm-embedded>



COLAController



RTX EAI Port Server

A més, a part d'Amelie SDK es necessari el GCC per la compilació del codi i la creació de l'arxiu .fws, que es el que fa servir COLAcontroller per a pujar l'aplicació

Desenvolupament

1. SoftAp

Principals tasques realitzades

- Pt_sotAP : Protothread que crea el fil d'execució de la tasca
- Pt_AppWifiReset: Inicialitza el Thread Wifi
- AppWifiIpV4Config(): Configura aspectes com la direcció i la mascara
- PT_WAIT_UNTIL(Pt, IS_RECIEVED(API_WIFI_CONNECT_IND));: Esperar connexió client.

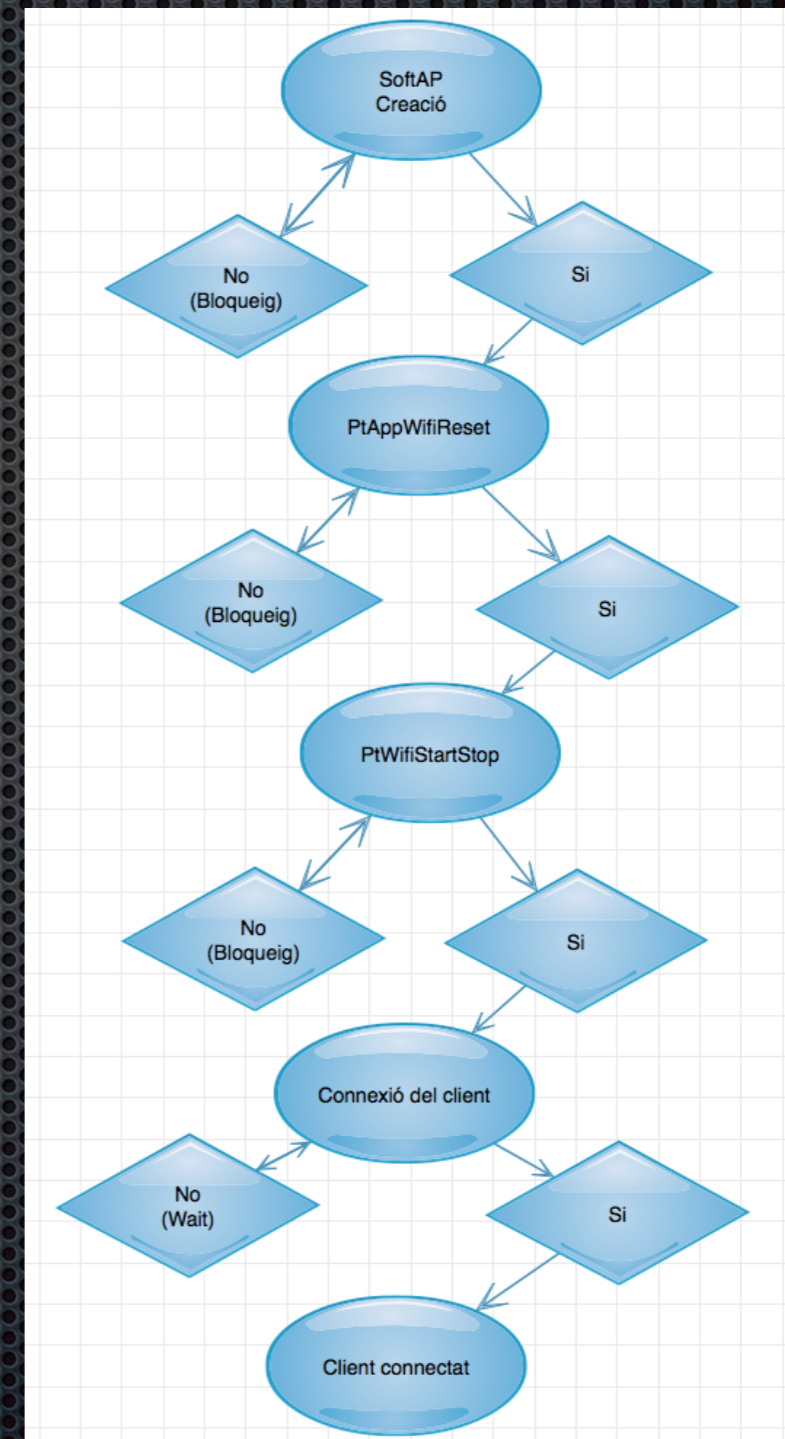


Diagrama SOFTAP

Desenvolupament

2.SPI

Principals tasques realitzades

RTX4100 (ROL "SLAVE")

- Creació Bus inici
- Espera de la Rebuda de dades del SCDVK
- Una vegada rebudes, s'envien a la "struct" amb els diferents paràmetres.

ARDUINO (ROL "MASTER")

- Inicialització del bus SPI i posar-lo a nivell "HIGH"
- Inicialitzar l'obtenció de temperatura i humitat al header SCDKV (getTemperature(), getHumidity())
- Activar l'escriptura de dades
- Definir la configuració de la comunicació SPI (SPISettings)
- Escriure al bus per enviar les dades amb SPI.transfer

```
COM3 (Arduino Due (Programming Port))
* Is your wiring correct?
* did you change the chipSelect pin to match your shield or module?
Prova enviament:
->Enviament Temperatura: 31.86
->Enviament Humitat: 41.67
->Enviament de control: 4
->Enviament control millis: 19379
Prova dades rebudes:
<-Rebu1: 31
<-Rebu2: 41
<-Rebu2: 3
Temperature: 31.85 C, Humidity: 41.68 %
```

SPI Communication

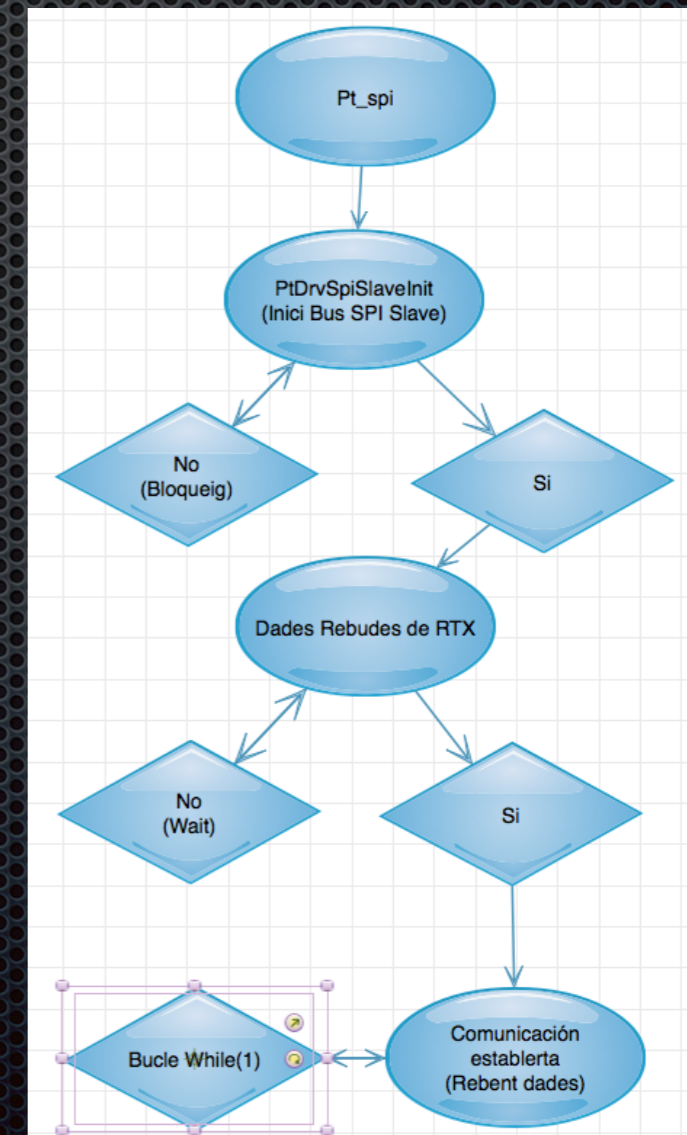


Diagrama SPI Communication

Desenvolupament

3. WebServer

Generar la pàgina Web

- Creació de Capçalera Web(AddHeader)
- Generar la Web (GenerateMaingPage)

Inici esglaonat del sistema

- Crida al fill Pt_softAp: Inici AP
- Rescata la temperatura interna del RTX4100
- Inicia servidor i espera fins rebre resposta(SendApiHttpServerInitReq)
- Comprovar l'inici del servidor
- Crear pàgina al servidor
- Crida a fill Pt_Spi per obtenir dades

4. Resultats

- Comunicació SPI
- Rebuda de temperatura Interna RTX4100 i publicació al WebServer
- Temperatura del SCDVK rebuda per SPI però amb errades a la publicació
- Humitat del SCDVK rebuda per SPI però no es mostre bé
- Mostreig de control

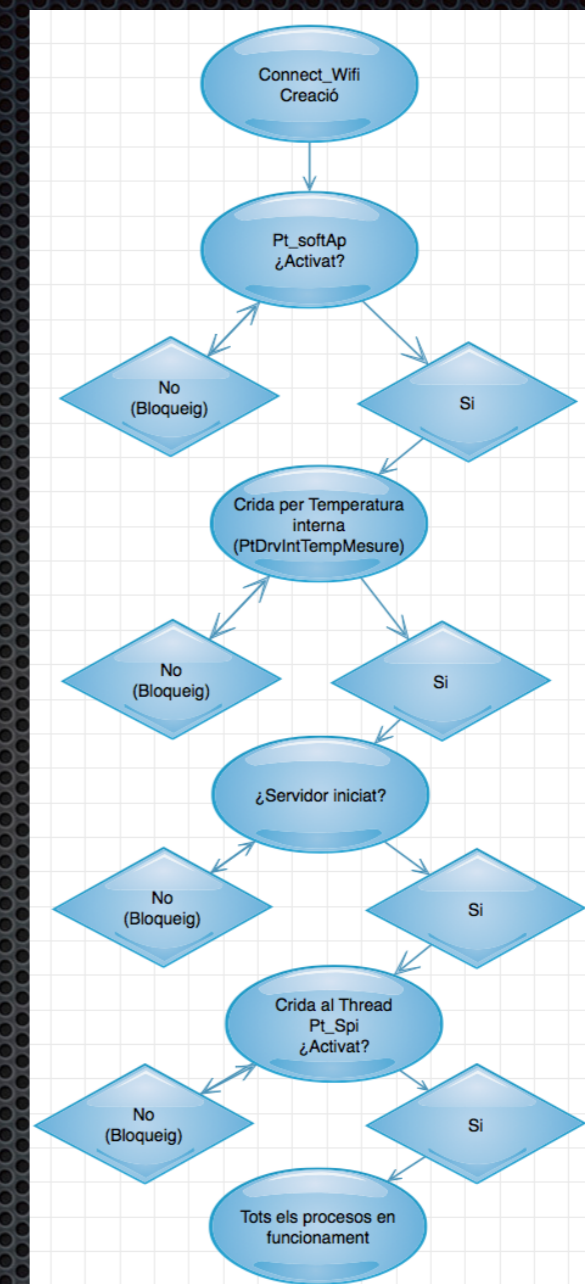


Diagrama SPI Communication

SERVER

Temperatura: 31

Humitat: 76

Mostreig nº 16

Temperatura RTX: 38.0 Graus

Entorn WebServer

✦ **Futures millores**

- Depuració i optimització del codi generat, tant el main.c com el Firmware SCDVK 1.5
- Resoldré els problemes en mostrar Temperatura i humitat a la pàgina Web del servidor
- Millorar l'enviament de comandes
- Inserció de formularis

✦ **Conclusions**

- Estudi de les eines, arxius i diferents qüestions teòriques, per la generació del codi proposat
- Implementació del mode SOFTAP
- Creació del servidor i la seva Web
- Disseny del sistema SPI per l'enviament de dades entre el SCDVK i el mòdul Wi-Fi RTX4100
- Establir una entesa amb el funcionament d'Arduino i l'edició del seu codi.