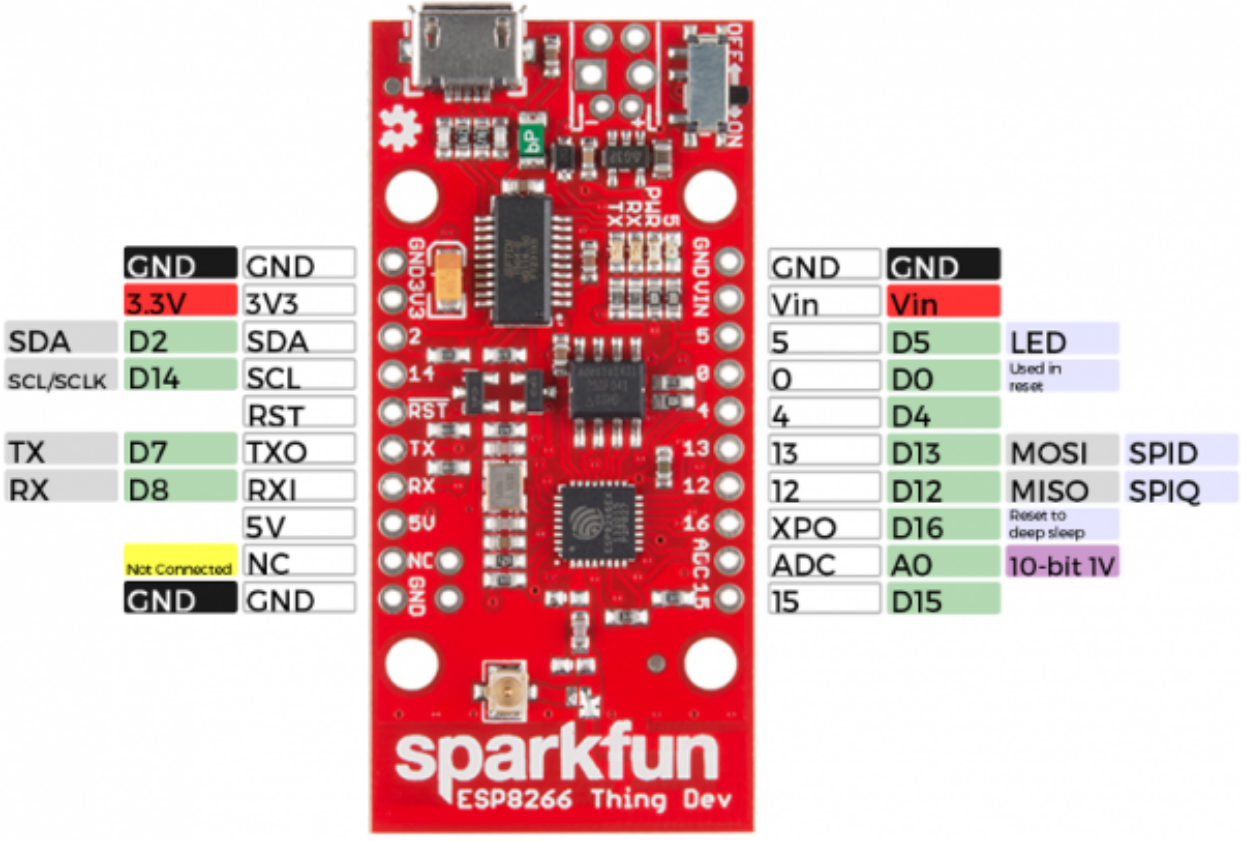
IoT (Internet of Things)

Servidor Web con ESP8266

Departamento de Ingeniería en Sistemas y Computación
Universidad Católica del Norte, Antofagasta.

ESP8266

- WiFi integrado
- ADC de 10 bits
- 11 pines digitales (algunos habilitados para comunicación)
- interfaces de comunicación: I^2C,UART,SPI



Servidor Web en ESP8266

- Es posible utilizar esta tarjeta en modo
  - estación WiFi
  - punto de acceso
- en modo `estación` es posible usar mDNS (multicast DNS) para que el servidor apunte al dominio `thing.local`
- `const char WiFiSSID[] = "nombre_red";`
- `const char WiFiPSK[] = "clave_red";`
- para modo AP:
  `const char WiFiAPPSK[] = "clave";`

Modo estación

Iniciar servidor:

```
WiFiServer server(80);

void setup(){
  initHardware();//establecer pines correspondientes
  connectWiFi(); //definido en estacion_esp.ino
  server.begin();
  setupMDNS(); //definido en estacion_esp.ino
}
```

Modo estación

- Conectar a una red: `WiFi.begin(ssid, password);`
- Chequear estado de conexión: `WiFi.status()`
- Parámetro definido en librería `ESP8266WiFi.h` para definir estado de conexión: `WL_CONNECTED`
- Obtener IP asignada: `WiFi.localIP()`

Modo estación

Es posible incluir múltiples redes conocidas para conectar a la que tenga una señal más fuerte.

- incluir librería `ESP8266WiFiMulti.h`
- agregar par red-clave:
  `wifiMulti.addAP('Miguel_cel', 'claseiot');`
- búsqueda de red disponible:
  `while (wifiMulti.run() != WL_CONNECTED){`
  `Serial.println('buscando');}`
- obtención del nombre de red conectada: `WiFi.SSID()`

Modo estación

Chequear si hay algún cliente conectado:

```
void loop(){
  WiFiClient client = server.available();
  if (!client){
    return;
  }
⋮
}
```

Modo estación

Leer solicitud (primera línea):

```
void loop(){
  ⋮
  String req = client.readStringUntil('\r');
  Serial.println(req);
  client.flush();
  ⋮
}
```

Ejemplo

- Implemente un divisor de tensión con una resistencia de $10K\Omega$ y una fotorresistencia, con la salida del divisor conectada al pin ADC del ESP8266.
- Cargue el código de ejemplo `estacion_esp.ino` en ESP8266.
- Ingrese al navegador web de algún dispositivo conectado a la misma red, y verifique el resultado de ingresar las siguientes URLs:
  - `thing.local/read`
  - `thing.local/led/0`
  - `thing.local/led/1`
- para actualizar automáticamente la página, descomentar:

```
s += "<meta http-equiv='refresh' content='1'/>\r\n";
```

Modo Access Point

- **no** se configura el mDNS
- `WiFi.mode(WIFI_AP);` en lugar de `WiFi.mode(WIFI_STA);`
- `WiFi.softAP(nombre, clave);`
- cargar código de ejemplo `accesspoint_esp.ino` e ingresar URLs anteriores (ingresando la IP en lugar de thing.local)
- esta vez el dispositivo cliente se debe conectar a la **red ESP8266 ThingDev-XXXX**

Actividad

Cree un sitio web (simple, puede ser en HTML) que tenga un botón para activar y otro botón para desactivar la salida de un pin digital del ESP8266.

- Primero conecte un LED a ese pin digital
- Luego reemplace el LED por un relé que intervenga la alimentación de una ampolleta de 220 volts.

  Precaución: realice las conexiones con la alimentación apagada y verifique que los terminales de alimentación no hacen contacto entre sí ni con ninguna superficie metálica.