

## 4 – Interconectando dos MCU

### 4a – Comunicación UDP básica

Creamos un esquemático nuevo en blanco. **File -> New (Ctrl + N)**

- 1) Incorporar los siguientes dispositivos y elementos:
  - 2 MCU: **Components -> Boards -> MCU Board**
  - Cafetera: **End Devices -> Home -> Appliance**
  - Interruptor: **Components -> Sensors -> Rocker Switch**
  - Hub: **Network Devices -> Hubs -> Hub-PT**
  
- 2) Editar los MCU y añadirles un módulo PT-IOT-NM-1CE:
  - Abrir cada **MCU -> Physical -> PT-IOT-NM-1CE** y arrastrar la placa hacia el hueco en la esquina inferior derecha de la placa MCU
  
- 3) Interconectarlos utilizando **Connections -> IoT Custom Cable** de la siguiente manera:
  - Interruptor Pin D0 -> MCU0 Pin D0
  - Cafetera Pin D0 -> MCU1 Pin D0
  
- 4) Interconectar con cable de red **Connections -> Cooper Straight Through** de los siguientes componentes:
  - MCU0.Ethernet0 -> Hub.FastEthernet0
  - MCU1.Ethernet0 -> Hub.FastEthernet1
  
- 5) Configurar las direcciones IP estáticas:
  - MCU0 -> Config -> Ethernet0 -> IP Configuration -> Static:
    - IP Address: 192.168.1.2
    - Subnet Mask: 255.255.255.0
  - MCU1 -> Config -> Ethernet0 -> IP Configuration -> Static:
    - IP Address: 192.168.1.1
    - Subnet Mask: 255.255.255.0
  
- 6) Programar los MCU para enviar y recibir los comandos de activación/desactivación de la cafetera:
  - Seleccionar MCU0 -> Programming:
    - Pulsar el botón New.
    - Nombre proyecto: CafeteraSend
    - Template: Empty – Javascript
    - Pulsar el botón Create.
    - Doble click sobre “main.js” y copiar el siguiente código:

```

var port = 1234;
var dstIP = "192.168.1.1";

var socket;
var state ;

function setup() {
  pinMode(0,INPUT);
  state = 0;

  socket = new UDPSocket();

  // Recepcion UDP
  socket.onReceive = function(ip, port, data) {
    Serial.println("Recibido de "
      + ip + ":" + port + ": " + data);
  };

  // Activa el socket UDP en el puerto
  Serial.println(socket.begin(port));
}

function loop() {
  if (digitalRead(0)) {
    if (state === 0) {
      state = 1;
      socket.send(dstIP, port, "1");
      Serial.println("ON");
    }
  }
  else{
    if (state === 1) {
      state = 0;
      socket.send(dstIP, port, "0");
      Serial.println("OFF");
    }
  }
  delay(1000);
}

```

- Seleccionar MCU1 -> Programming:
  - Pulsar el botón New.
  - Nombre proyecto: CafeteraRecv
  - Template: Empty – Javascript
  - Pulsar el botón Create.
  - Doble click sobre “main.js” y copiar el siguiente código:

```

var port = 1234;
var dstIP = "192.168.1.2";
var socket;

function setup() {
  socket = new UDPSocket();
  customWrite(0,"0");
}

```

```

// Recepcion
socket.onReceive = function(ip, port, data) {
    Serial.println("Recibido de "
        + ip + ":" + port + ": " + data);
    if(data=="1"){
        Serial.println("CAFETERA ENCENDIDA");
        customWrite(0,"1");
    }
    else {
        Serial.println("CAFETERA APAGADA");
        customWrite(0,"0");
    }
};

// Activa el socket UDP en el puerto
Serial.println(socket.begin(port));
}

function loop() {
    // Nada
}

```

## Resultado de la Simulación

Active el pulsador, pulsando la tecla ALT y clickando sobre el pulsador. Podrá observar cómo se enciende la cafetera. Desactive el pulsador (ALT + Click sobre el pulsador). La cafetera se apagará. Observando el log de cada MCU podrá ver los mensajes enviados y recibidos.

## 4b – Comunicación UDP con múltiples destinos

Crear un diseño con 4 MCU (**Components -> Boards -> MCU Board**), 3 pulsadores (**Components -> Sensors -> Rocker Switch**), 1 lámpara (**End Devices -> Home -> Light**), 1 humidificador (**End Devices -> Home -> Humidifier**) y 1 cafetera (**End Devices -> Home -> Appliance**). El primer MCU tendrá conectada una lámpara. El segundo MCU tendrá conectado un humidificador. El tercer MCU tendrá conectado la cafetera. El último MCU tendrá 3 botones pulsadores que seleccionarán respectivamente la activación/desactivación de la lámpara, el humidificador o la cafetera.

Los MCU se interconectarán entre sí mediante un cable Ethernet (**Connections -> Cooper Straight Through**) conectados a un Hub Ethernet (**Network Devices -> Hubs -> Hub-PT**).

Configurar convenientemente la tarjeta Ethernet de cada MCU, asignándole la IP conveniente a cada uno.

Incorporar el código correspondiente en cada MCU para realizar la tarea solicitada.

#### 4c – Comunicación UDP con múltiples fuentes a un mismo destino (múltiples puertos en destino)

Crear un diseño con 4 MCU (**Components -> Boards -> MCU Board**), 3 pulsadores (**Components -> Sensors -> Rocker Switch**), 1 lámpara (**End Devices -> Home -> Light**), 1 humidificador (**End Devices -> Home -> Humidifier**) y 1 cafetera (**End Devices -> Home -> Appliance**). El primer MCU (MCU0) tendrá conectada una lámpara, un humidificador y la cafetera. El segundo MCU (MCU1) tendrá conectado un pulsador. El tercer MCU (MCU2) tendrá conectado otro pulsador. El último MCU (MCU3) tendrá otro botón pulsador. Cada uno de estos botones que seleccionarán respectivamente la activación/desactivación de la lámpara (asociado al MCU1), el humidificador (con el MCU2) o la cafetera (con el MCU3).

Los MCU se interconectarán entre sí mediante un cable Ethernet (**Connections -> Cooper Straight Through**) conectados a un Hub Ethernet (**Network Devices -> Hubs -> Hub-PT**).

Configurar convenientemente la tarjeta Ethernet de cada MCU, asignándole la IP conveniente a cada uno.

Hay dos opciones de implementación (implementar ambas y comparar que los resultados deben ser los mismos):

##### *4.c.1 Crear diferentes sockets asociados a diferentes puertos en el destino.*

Seleccionar el componente de destino asociándole un puerto diferente (crear 3 sockets diferentes, cada uno en un puerto distinto).

##### *4.c.2 Crear un protocolo de selección incorporado en el propio flujo de datos*

Crear un protocolo que incorpore en el payload del paquete enviado la selección del componente y el comando de activación o desactivación (por ejemplo, incluyendo un carácter asociado al componente: 'L' para la lámpara; 'H' para el humidificador; y 'C' para la cafetera, seguido del valor de activación o desactivación, según corresponda).

Incorporar el código correspondiente en cada MCU para realizar la tarea solicitada.