ESCUELA POLITÉCNICA SUPERIOR DE CÓRDOBA
Universidad de Córdoba

EPSC

Internet de las Cosas

Máster Universitario en Inteligencia Computacional e Internet de las Cosas

Práctica 2. Interconectando dos MCU

Carlos Checa Moreno

UNIVERSIDAD D CÓRDOBA

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

Práctica 2

INTERCONECTANDO DOS MCU

2a – 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**

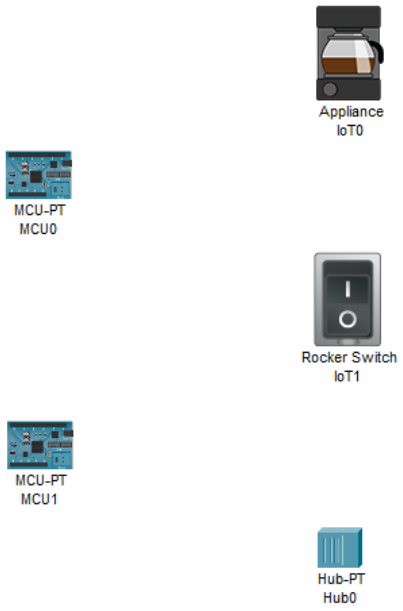

Figura 1: Dispositivos colocados

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

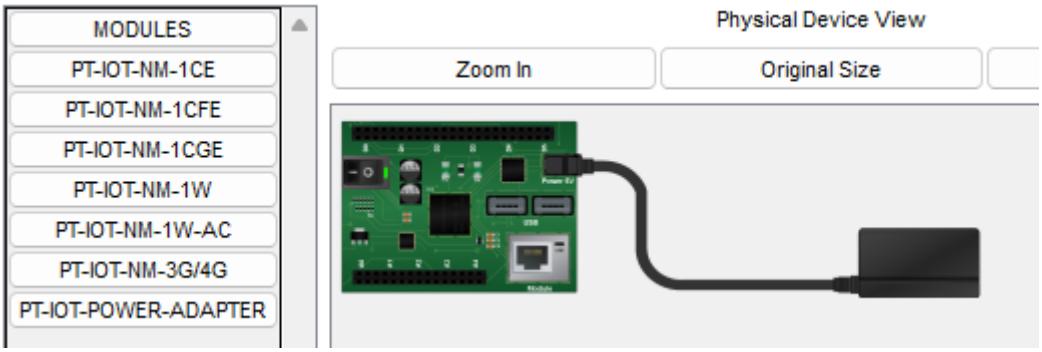

Figura 2: Módulos añadidos

3) Interconectarlos utilizando **Connections -> IoT Custom Cable** de la siguiente manera:

   - Interruptor Pin D0 -> MCU0 Pin D0
   - Cafetera Pin D0 -> MCU1 Pin D0

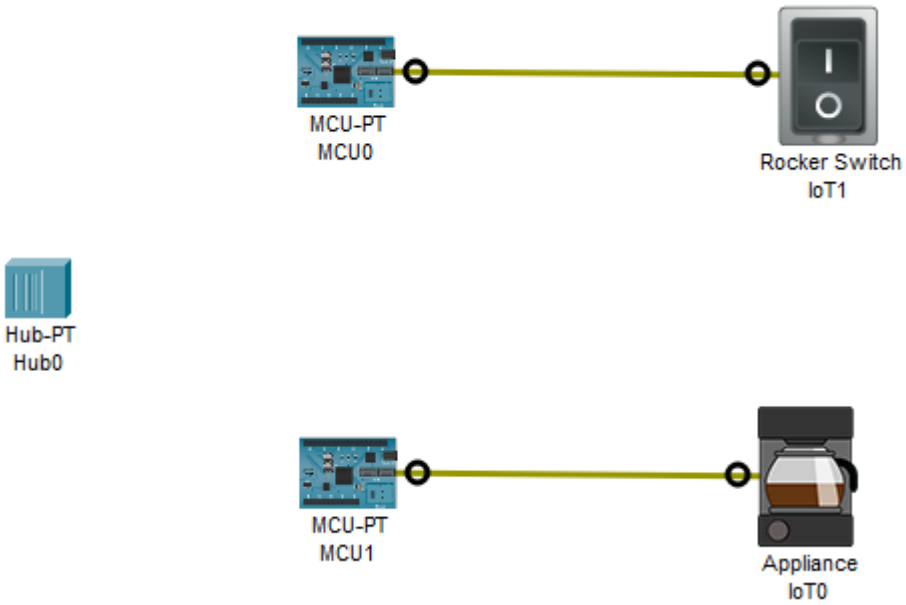

Figura 3: Conexiones IoT

4) Interconectar con cable de red **Connections -> Cooper Straight Through** de los siguientes componentes:

- MCU0.Ethernet0 -> Hub.FastEthernet0
- MCU1.Ethernet0 -> Hub.FastEthernet1

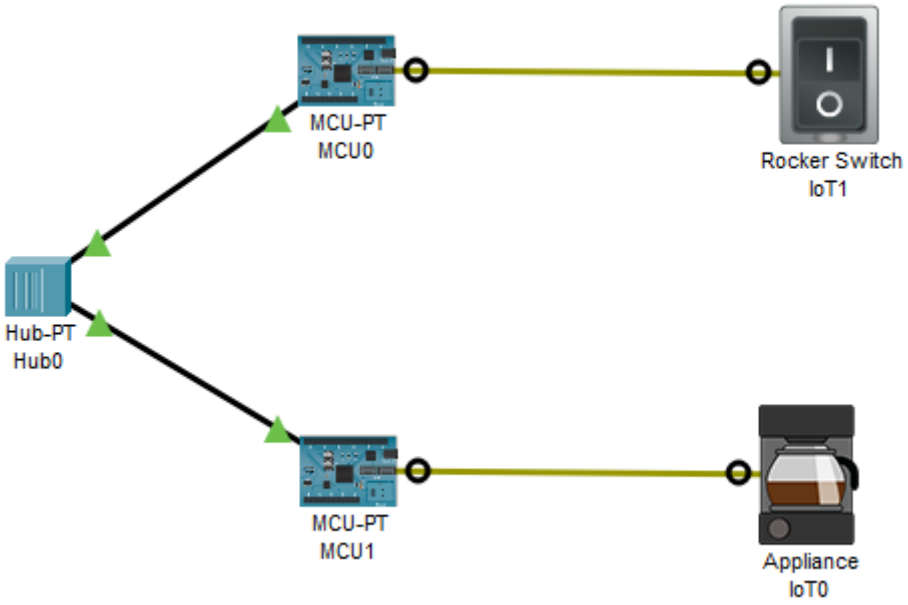

Figura 4: Conexiones Ethernet

5) Configurar las direcciones IP estáticas:

- MCU0 -> Config -> Ethernet0 -> IP Configuration -> Static:
  - IP Address: 192.168.1.2
  - Subnet Mask: 255.255.255.0

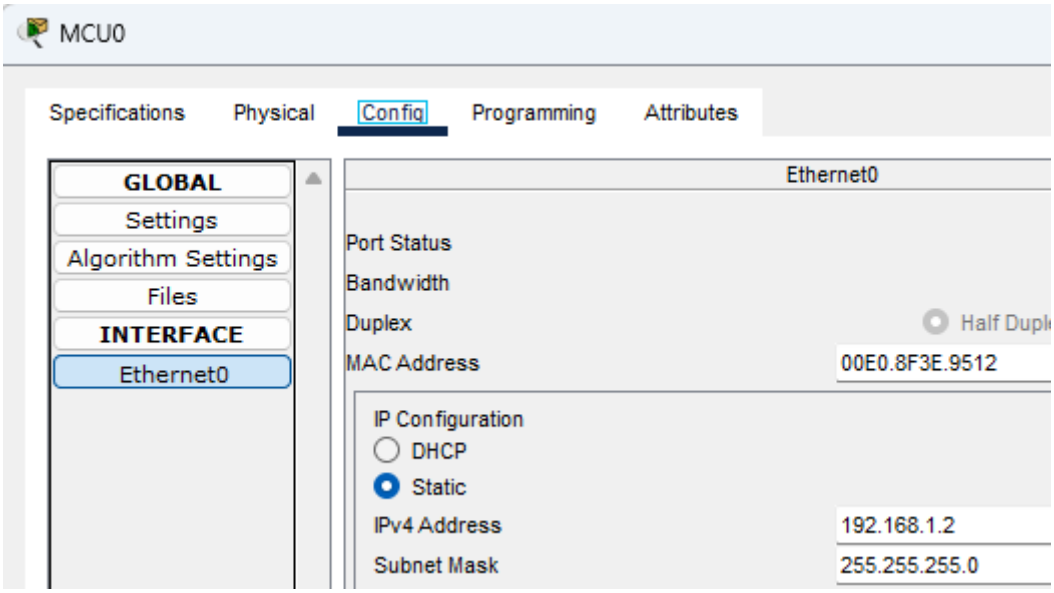

Figura 5: IP MCU0

- MCU1 -> Config -> Ethernet0 -> IP Configuration -> Static:
    - IP Address: 192.168.1.1
    - Subnet Mask: 255.255.255.0

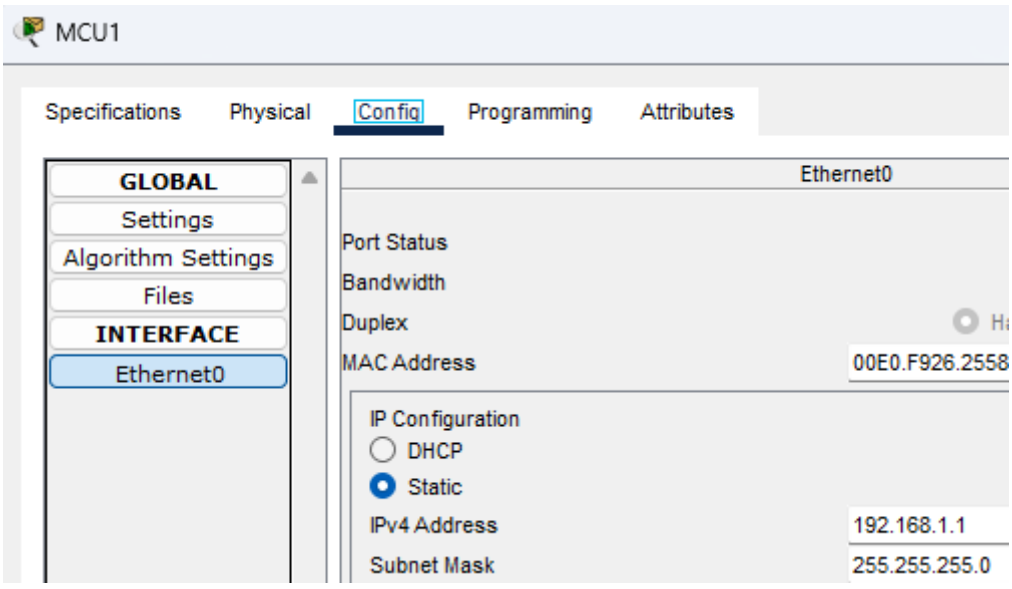

Figura 6: IP MCU1

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

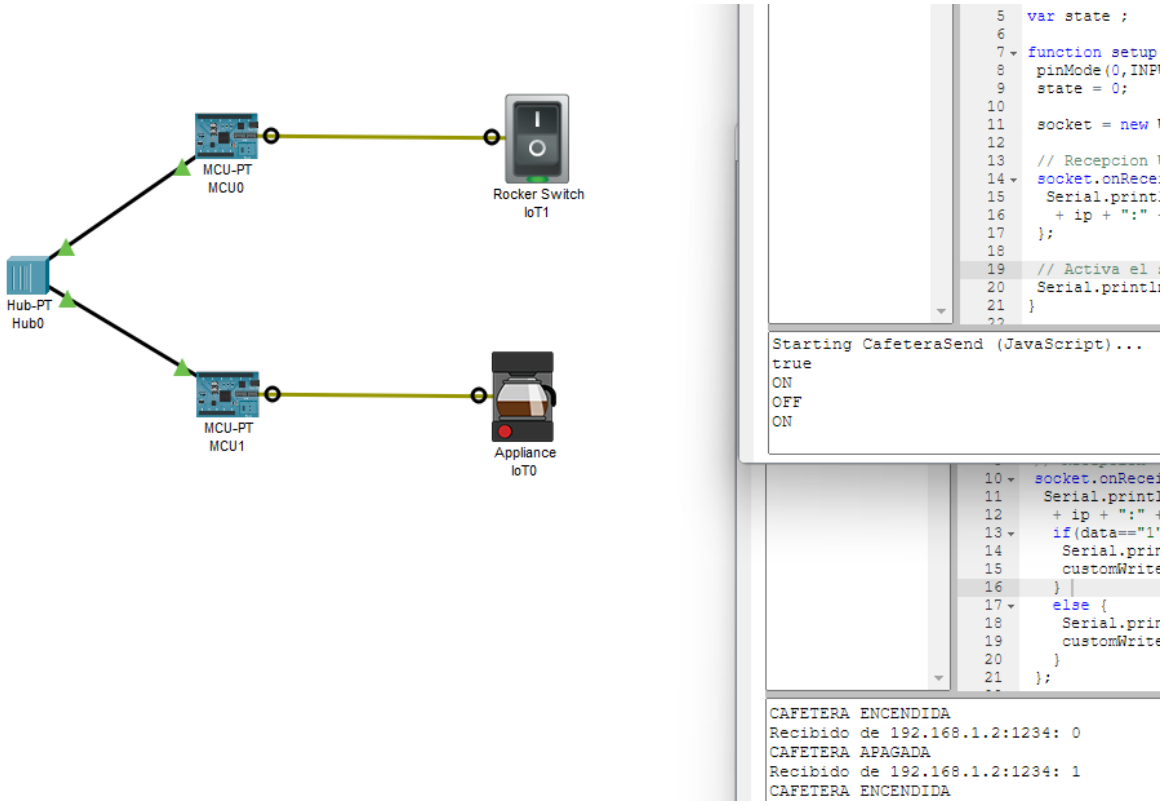

Figura 7: Resultado simulación

2b – 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.

Como se indica en el ejercicio, he utilizado 4 placas MCU interconectadas mediante un hub Ethernet utilizando cables Ethernet de tipo "Cooper Straight Through". Al MCU1 conecté una lámpara, al 2 un humidificador y al 3 una cafetera. El MCU0 se configuró con tres pulsadores, cada uno asignado a controlar la activación/desactivación de un dispositivo específico (lámpara, humidificador o cafetera).

Para establecer una comunicación adecuada entre los MCU y los dispositivos finales a través del protocolo UDP ha sido necesario asignar IPs a las 4 placas siendo MCU1: 192.168.1.1; MCU2: 192.168.1.2; MCU3: 192.168.1.3 y MCU0: 192.168.1.4.

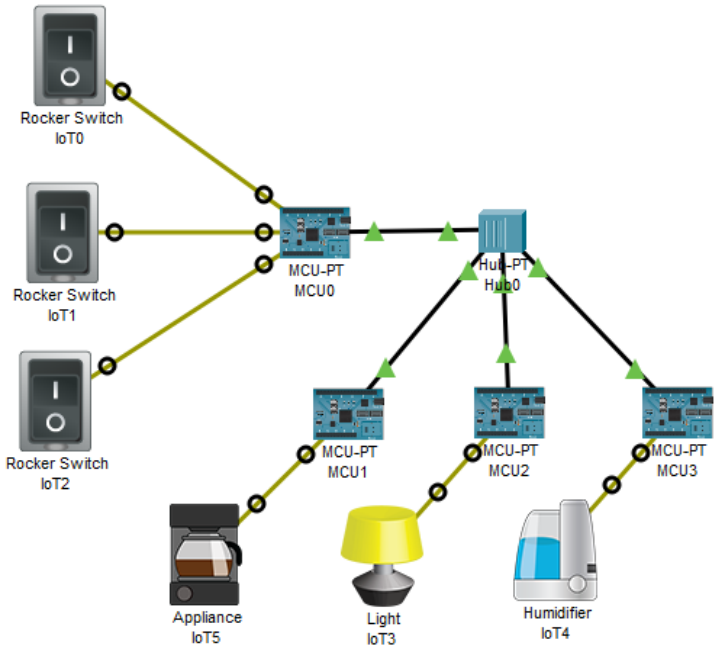

Figura 8: Simulación. Todo apagado

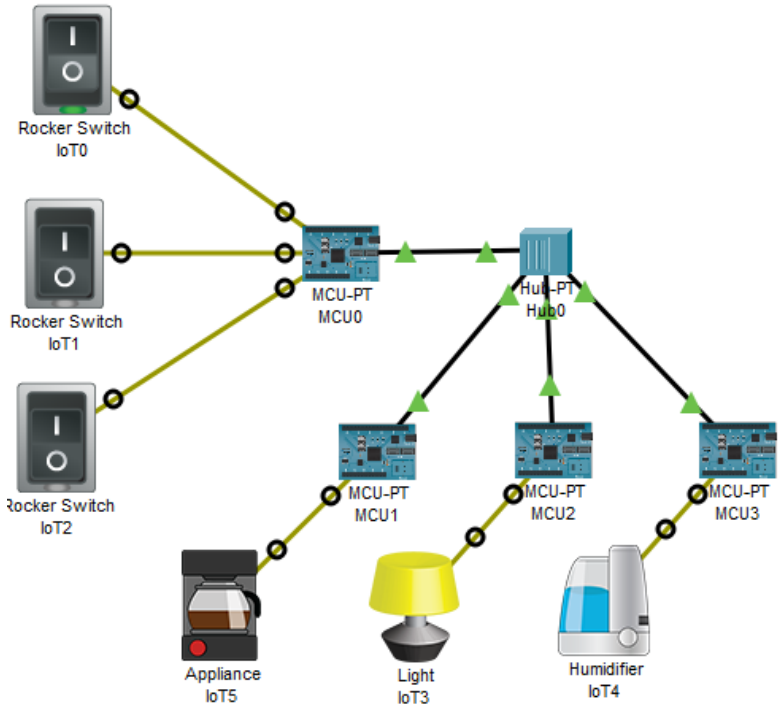

Figura 9: Simulación. Cafetera activa

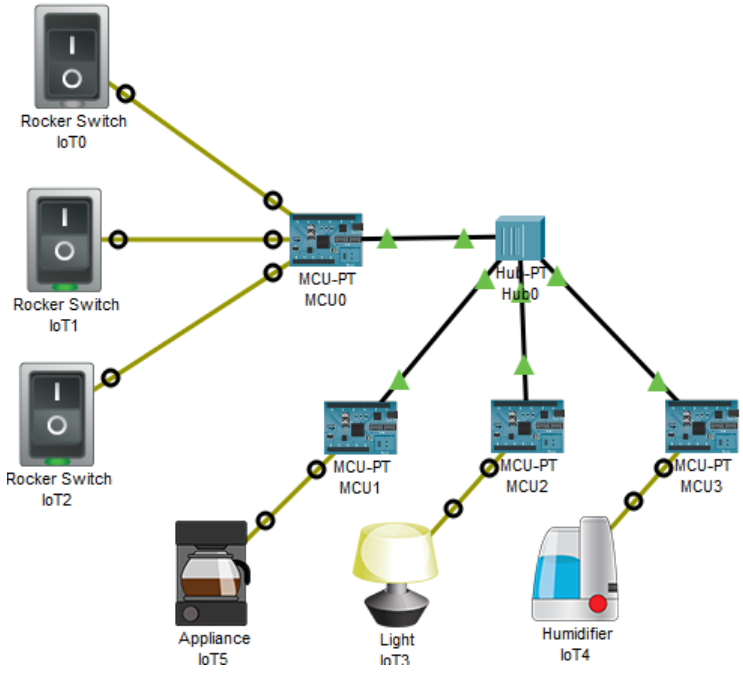

Figura 10: Simulación. Lámpara y humificador activos

Código MCU0:

```
var port = 1234;
var dstIP_Cafetera = "192.168.1.1";
var dstIP_Lampara = "192.168.1.2";
var dstIP_Humidificador = "192.168.1.3";
var socket;
var state_Cafetera = 0;
var state_Humidificador = 0;
var state_Lampara = 0;

function setup() {

    pinMode(0, INPUT);
    pinMode(1, INPUT);
    pinMode(2, INPUT);

    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() {

    // Cafetera
    if (!digitalRead(0)) {
        if (state_Cafetera === 1)
        {
            state_Cafetera = 0;
            socket.send(dstIP_Cafetera, port, "0");
            Serial.println("CAFETERA OFF");
        }
    } else {
        if (state_Cafetera === 0) {
            state_Cafetera = 1;
            socket.send(dstIP_Cafetera, port, "1");
            Serial.println("CAFETERA ON");
        }
    }

    // Lampara
    if (digitalRead(1)) {
        if (state_Lampara === 0)
        {
            state_Lampara = 1;
            socket.send(dstIP_Lampara, port, "1");
            Serial.println("LAMPARA ON");
        }
    } else {
        if (state_Lampara === 1)
        {
            state_Lampara = 0;
```

```
            socket.send(dstIP_Lampara, port, "0");
            Serial.println("LAMPARA OFF");
        }
    }

    // Humidificador
    if (digitalRead(2))
    {
        if (state_Humidificador === 0)
        {
            state_Humidificador = 1;
            socket.send(dstIP_Humidificador, port, "1");
            Serial.println("HUMIDIFICADOR ON");
        }
    } else {
        if (state_Humidificador === 1)
        {
            state_Humidificador = 0;
            socket.send(dstIP_Humidificador, port, "0");
            Serial.println("HUMIDIFICADOR OFF");
        }
    }

    delay(100);
}
```

Código MCU1 Cafetera:

```
var port = 1234;
var socket;

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

      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");
            }
      };

      Serial.println(socket.begin(port));
}

function loop()
{
// Nada
}
```

Código MCU2 lámpara:

```
var port = 1234;
var socket;

function setup()
{
	socket = new UDPSocket();
	customWrite(0,"0");
	socket.onReceive = function(ip, port, data)
	{
		Serial.println("Recibido de " + ip + ":" + port + ": " + data);
		if(data=="1")
		{
			Serial.println("LAMPARA ENCENDIDA");
			customWrite(0,"2");
		}
		else
		{
			Serial.println("LAMPARA APAGADA");
			customWrite(0,"0");
		}
	};

	Serial.println(socket.begin(port));
}

function loop()
{
// Nada
}
```

Código MCU3 humificador:

```
var port = 1234;
var socket;

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

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

	Serial.println(socket.begin(port));
}
```

```
function loop()
{
// Nada
}
```

2c – 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):

2c.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).

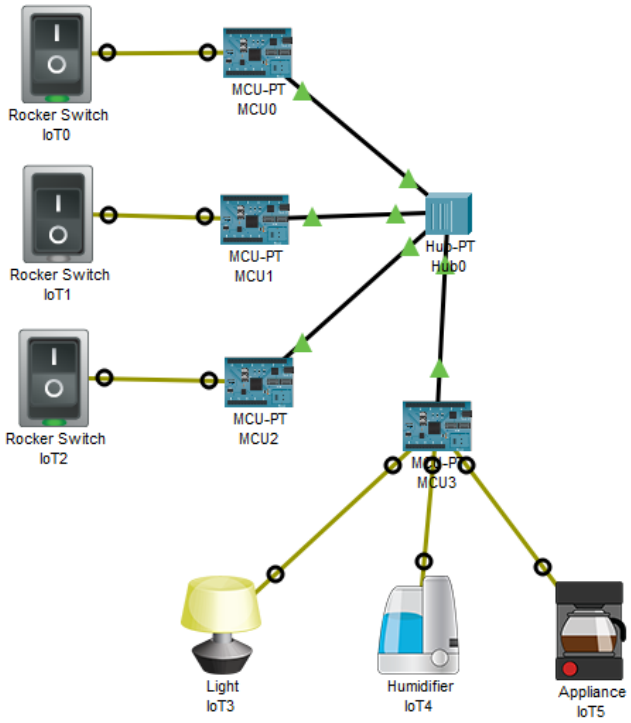

Figura 11: Simulación con 3 sockets

Para esta configuración debemos programar la comunicación en la placa MCU3. Los interruptores tienen diferentes puertos asignados, desde el 1234 al 1236. Aprovechando que tenemos 3 puertos diferentes se creará un socket para cada uno de estos puertos para enlazar los interruptores con sus correspondientes dispositivos.

Código MCU3:

```
var port_Lampara = 1234;
var port_Humidificador = 1235;
var port_Cafetera = 1236;

var dstIP_Lampara = "192.168.1.2";
var dstIP_Humidificador = "192.168.1.3";
var dstIP_Cafetera = "192.168.1.4";

var socket_Lampara;
var socket_Humificador;
var socket_Cafetera;

function setup() {
      socket_Lampara = new UDPSocket();
      customWrite(0,"0");
      socket_Lampara.onReceive = function(ip, port_Lampara, data)
      {
            if (ip == dstIP_Lampara)
            {
                  Serial.println("Recibido de "
                  + ip + ":" + port_Lampara + ": " + data);
                  if(data=="1")
                  {
                        Serial.println("LAMPARA ENCENDIDA");
                        customWrite(2,"1");
                  }
                  else
                  {
                        Serial.println("LAMPARA APAGADA");
                        customWrite(2,"0");
                  }
            }
      };
      // Activa el socket UDP en el puerto
      Serial.println(socket_Lampara.begin(port_Lampara));

      socket_Humificador = new UDPSocket();
      socket_Humificador.onReceive = function(ip, port_Humidificador, data)
      {
            if (ip == dstIP_Humidificador)
            {
                  Serial.println("Recibido de "
                  + ip + ":" + port_Humidificador + ": " + data);
                  if(data=="1")
                  {
                        Serial.println("HUMIDIFICADOR ENCENDIDA");
                        customWrite(0,"1");
                  }
                  else
                  {
                        Serial.println("HUMIDIFICADOR APAGADA");
                        customWrite(0,"0");
                  }
            }
      };
      // Activa el socket UDP en el puerto
      Serial.println(socket_Humificador.begin(port_Humidificador));
```

```
	socket_Cafetera = new UDPSocket();
	socket_Cafetera.onReceive = function(ip, port_Cafetera, data)
	{
		if (ip == dstIP_Cafetera)
		{
			Serial.println("Recibido de "
			+ ip + ":" + port_Cafetera + ": " + data);
			if(data=="1")
			{
				Serial.println("CAFETERA ENCENDIDA");
				customWrite(1,"1");
			}
			else
			{
				Serial.println("CAFETERA APAGADA");
				customWrite(1,"0");
			}
		}
	};
	// Activa el socket UDP en el puerto
	Serial.println(socket_Cafetera.begin(port_Cafetera));

}

function loop()
{
// Nada
}
```

Código MCU1 Lampara:

```
// Lampara
var port = 1234;
var dstIP = "192.168.1.1";
var socket;
var state;

function setup()
{
	pinMode(2,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("LAMPARA ON");
```

```
            }
      }
      else
      {
            if (state === 1)
            {
                  state = 0;
                  socket.send(dstIP, port, "0");
                  Serial.println("LAMPARA OFF");
            }
      }
}
```

Código MCU2 Humificador:

```
// Humidificador
var port = 1235;
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("HUMIDIFICADOR ON");
            }
      }
      else
      {
            if (state === 1)
            {
                  state = 0;
                  socket.send(dstIP, port, "0");
                  Serial.println("HUMIDIFICADOR OFF");
            }
      }
}
```

Código MCU3 Cafetera:

```
// Cafetera
var port = 1236;
var dstIP = "192.168.1.1";
var socket;
var state;

function setup()
{
      pinMode(1,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("CAFETERA ON");
            }
      }
      else
      {
            if (state === 1)
            {
                  state = 0;
                  socket.send(dstIP, port, "0");
                  Serial.println("CAFETERA OFF");
            }
      }
}
```

2c.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.

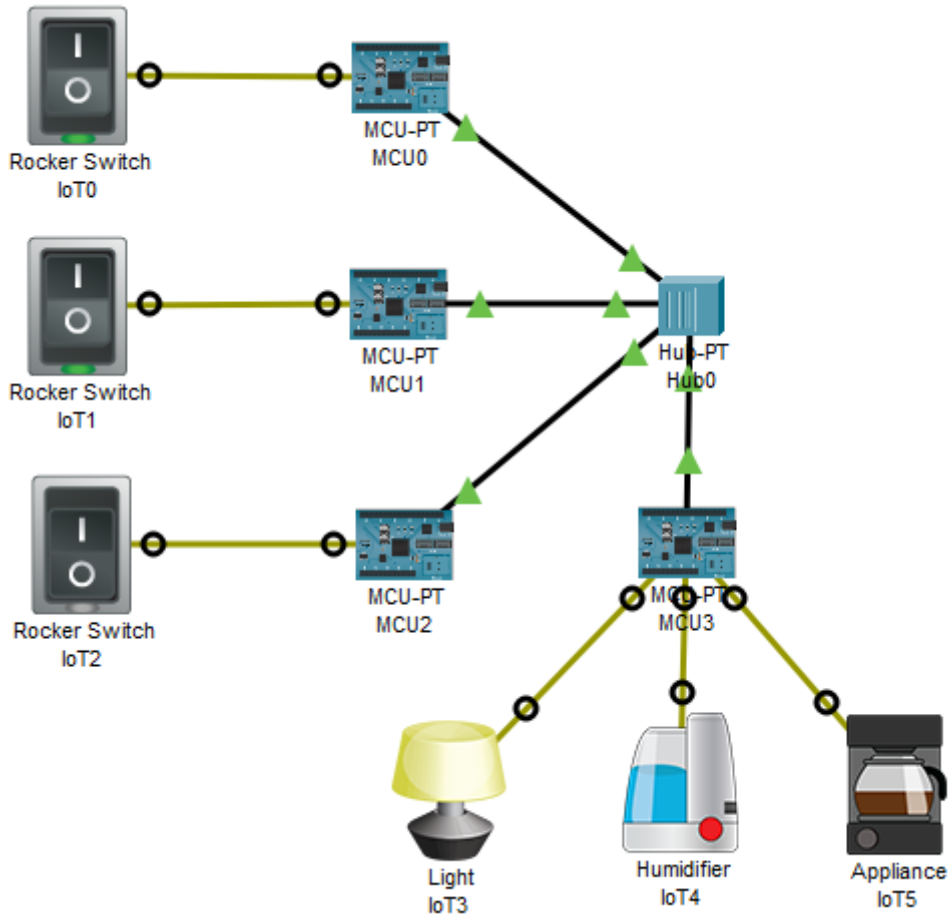

Figura 12: Simulación protocolo de selección

En este caso, en vez de crear 3 sockets como se hizo anteriormente, se ha desarrollado un protocolo que analiza los mensajes que se envían por las placas conectadas a los interruptores. En “data = data.split(" ");” dividimos la letra que indica a que dispositivo va dirigido el mensaje y el número que indica el estado de este. Con esta información ya podemos saber el qué se quiere apagar o encender.

Código MCU3:

```
var port = 1234;
var dstIP_Lampara = "192.168.1.2";
var dstIP_Humidificador = "192.168.1.3";
var dstIP_Cafetera = "192.168.1.4";
var socket;

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

      // Recepcion
      socket.onReceive = function(ip, port, data)
      {
            data = data.split(" ");
            if (data[0] == "H")
            {
                  Serial.println("Recibido de "
                  + ip + ":" + port + ": " + data);
                  if(data[1]=="1")
                  {
                        Serial.println("HUMIDIFICADOR ENCENDIDA");
                        customWrite(0,"1");
                  }
                  else
                  {
                        Serial.println("HUMIDIFICADOR APAGADA");
                        customWrite(0,"0");
                  }
            }
            else if (data[0] == "L")
            {
                  Serial.println("Recibido de "
                  + ip + ":" + port + ": " + data);
                  if(data[1]=="1")
                  {
                        Serial.println("LAMPARA ENCENDIDA");
                        customWrite(2,"1");
                  }
                  else
                  {
                        Serial.println("LAMPARA APAGADA");
                        customWrite(2,"0");
                  }
            }
            else if (data[0] == "C")
            {
                  Serial.println("Recibido de "
                  + ip + ":" + port + ": " + data);
                  if(data[1]=="1")
                  {
                        Serial.println("CAFETERA ENCENDIDA");
                        customWrite(1,"1");
                  }
                  else {
                        Serial.println("CAFETERA APAGADA");
                        customWrite(1,"0");
                  }
```

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

function loop()
{
// Nada
}
```

Código MCU0 Lámpara:

```
// Lampara
var port = 1234;
var dstIP = "192.168.1.1";
var socket;
var state;

function setup()
{
      pinMode(2,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()
{
      // LAMPARA
      if (digitalRead(0))
      {
            if (state === 0)
            {
                  state = 1;
                  socket.send(dstIP, port, "L 1");
                  Serial.println("LAMPARA ON");
            }
      }
      else
      {
            if (state === 1)
            {
                  state = 0;
                  socket.send(dstIP, port, "L 0");
                  Serial.println("LAMPARA OFF");
            }
      }
}
```

Código MCU1 Humificador:

```
// Humidificador
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, "H 1");
                  Serial.println("HUMIDIFICADOR ON");
            }
      }
      else{
            if (state === 1)
            {
                  state = 0;
                  socket.send(dstIP, port, "H 0");
                  Serial.println("HUMIDIFICADOR OFF");
            }
      }
}
```

Código MCU2 Cafetera:

```
// Cafetera
var port = 1234;
var dstIP = "192.168.1.1";
var socket;
var state;

function setup()
{
      pinMode(1,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()
{
      // CAFETERA
      if (digitalRead(0))
      {
            if (state === 0)
            {
                  state = 1;
                  socket.send(dstIP, port, "C 1");
                  Serial.println("CAFETERA ON");
            }
      }
      else
      {
            if (state === 1)
            {
                  state = 0;
                  socket.send(dstIP, port, "C 0");
                  Serial.println("CAFETERA OFF");
            }
      }
}
```