

## Protocolo de comunicación

Para la comunicación entre los distintos MCUs se optó por el protocolo NEC a 38 kHz, dado que es un estándar ampliamente utilizado en sistemas de transmisión por infrarrojo.

### Protocolo NEC 38Khz:



- 8bits para direccionamiento
- 8bits para comando

De esta manera, se pueden representar hasta 256 valores distintos. En las siguientes secciones se detalla la estructura del protocolo propuesto, los comandos definidos y los distintos casos de uso (controlador  $\leftrightarrow$  sensor, controlador  $\leftrightarrow$  actuador y controlador en modo retransmisión).

A continuación detallamos los comandos a utilizar para la comunicación entre MCUs:

Comando	Valor
Identificador	0x00
Ack	0xFF
Temperatura_Req	0x10
Humedad_Req	0x11
Luminosidad_Req	0x12
STATUS	0x20
ON	0x21
OFF	0x22
Send_T	0x23
Send_H	0x24
Send_I	0x25

Table 1: Tabla de comandos

## Identificación de MCUs

El MCU controlador tiene una address (ej: 0x00) conocida por el resto de los dispositivos pasivos. El controlador enviará un mensaje a cada MCU pasivo para que éste envíe su address utilizando el comando Identificador (0x00). Luego, el dispositivo pasivo enviará el comando Identificador y el controlador obtendrá su address por el campo address.

El controlador enviará el comando identificador de a un dispositivo pasivo a la vez y lo solicitará hasta que lo reciba. No es necesario que el pasivo envíe un ACK porque enviará su comando de identificación todas las veces que se lo soliciten.

Se supone que los dispositivos pasivos:

- Saben la address del controlador.
- Tienen todos un address distinta.

## **MCU controlador ↔ MCU sensor**

En este escenario, el controlador solicita datos al sensor, como temperatura, humedad o luminosidad.

Características:

- Sin ACK: no es necesario que el sensor confirme cada solicitud con un mensaje separado, porque el controlador puede reintentar periódicamente hasta obtener una respuesta.
- Detección de error: si no llega respuesta después de un tiempo definido (timeout), el controlador asume que hay un problema de comunicación y lo reporta.

### **Mensajes**

- CONTROLADOR -> SENSOR
  - [addr\_control][T/H/L REQ]
- SENSOR -> CONTROLADOR
  - [addr\_sens][T/H/L RES]

## **MCU controlador ↔ MCU actuador**

### **Solicitud de estado**

Esta comunicación es realizada desde el controlador hacia el actuador para solicitar el estado de los actuadores. El controlador envía requests (STATUS) y el actuador responde con su estado actual (ON/OFF). Al igual que

### **Mensajes**

- CONTROLADOR -> ACTUADOR
  - [addr\_control][STATUS REQ]
- ACTUADOR -> CONTROLADOR
  - [addr\_act][ON/OFF RES]

### **Retransmisión de sensado**

En este caso si va a ser necesario implementar un control de recepcion, ya que, el controlador le solicitará al actuador para enviarle los datos ambientales.

### **Pasos:**

1. El controlador envía un mensaje indicando qué dato va a enviar (SEND\_T/H/L REQ). Esto permite que el actuador se prepare para recibir datos específicos.
2. El actuador confirma la recepción del mensaje de aviso (SEND\_ACK RES).
3. El controlador envía finalmente los datos ambientales (T/H/L RES).

**Diferenciación de mensajes:** para no confundir comandos con datos, el controlador utiliza direcciones distintas:

- 0x00 → comandos
- 0x01 → datos

**Casos de fallo:** Si el controlador no recibe ACK, reintenta el request un tiempo pre-seteado (timeout). Si el actuador no recibe los datos luego de enviar el ACK, seguirá enviando ACKs un tiempo pre-seteado (timeout). El controlador enviará los datos por cada vez que reciba un ACK del actuador.

### **Mensajes**

- CONTROLADOR (retransmisor) -> ACTUADOR
  - 1° [addr\_control][SEND\_T\_H\_L REQ]
  - 3° [addr\_control][T/H/L RES]
- ACTUADOR -> CONTROLADOR
  - 2° [addr\_act][ACK RES]