

Práctica 3: Recogida de medidas de temperatura, humedad y luminosidad con nodos TelosB

Objetivo:

El objetivo de la práctica es ampliar los programas desarrollados en la práctica 2 (esquema maestro-esclavo) para recoger medidas de los diferentes sensores de los nodos TelosB. El nodo maestro contendrá un programa que solicite lecturas de temperatura, humedad o luminosidad del nodo esclavo. Periódicamente enviará mensajes Zigbee codificando el ID del nodo origen, ID del destino y el tipo de lectura que desea. El nodo esclavo escuchará el canal, y al recibir un paquete sin errores de su nodo maestro, debe tomar la medida que se le indica y enviársela al maestro.

En cada mensaje que envíe el esclavo al maestro deberá tener los siguientes campos:

- ID del esclavo.
- RSSI del mensaje de petición de medida recibido.
- Tipo de medida.
- Medida solicitada.

Se utilizará un nodo base que escuche el canal y muestre por pantalla el intercambio de mensajes.

Recomendaciones:

- Utilizar los LEDs para representar la transmisión de los paquetes de diferente tipo.
- Ponerse de acuerdo con los compañeros para evitar solapamientos en el uso del canal y de los ID.
- Elegir los temporizadores para que permitan la adecuada visualización del experimento.

Se pide desarrollar los programas en TinyOS adecuados para este problema.

Notas:

Añadir las siguientes líneas a los archivos correspondientes:

Makefile

```
PFLAGS += -I$(TOSDIR)/platforms/telosa/chips/s1087
PFLAGS += -I$(TOSDIR)/platforms/telosa/chips/s10871
PFLAGS += -I$(TOSDIR)/platforms/telosa/chips/sht11
```

...AppC

```
components new SensirionSht11C() as Sht11;
components new HamamatsuS10871TsrC() as PhotoActiveC;
components new HamamatsuS1087ParC() as TotalSolarC;
```

```
App.ReadNotVisible -> PhotoActiveC;  
App.ReadVisible -> TotalSolarC;  
App.Temperature -> Sht11.Temperature;  
App.Humidity -> Sht11.Humidity;
```

...C

```
uses interface Read<uint16_t> as ReadVisible;  
uses interface Read<uint16_t> as ReadNotVisible;  
uses interface Read<uint16_t> as Temperature;  
uses interface Read<uint16_t> as Humidity;
```

```
uint16_t H;
```

```
call Humidity.read();
```

```
event void Humidity.readDone(error_t result, uint16_t val){  
    H=val;  
}
```

Datasheet del sensor de temperatura y humedad:

http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT1x_Datasheet_V5.pdf