**Project Report**

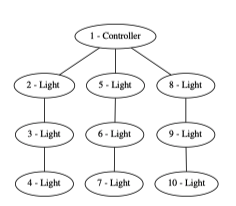
Daniele Mammone (10625264) - Gianmarco Naro (10610374)

Introduction

The aim of the project is to implement a Smart Network of Light Bulbs. This network is composed by a controller, and by smart bulbs. Even if they are all implemented on the same type of mote running tinyOS (Sky Mote type), the two categories of devices runs binaries generated from different codes. This allows to make lighter the binary for bulbs, and left to the controller only the parts necessary to manage the lights turning on and off.

Addressing

The addressing used is the same proposed by the requirements document. Here the scheme

just for reference.

Routing

In our routing strategy, two cases must be take in consideration:

* Command messages: when N receives a packet for X, with N ≠ X, if X > N the packet is redirected to X + 1, otherwise to X – 1. We didn’t take care of the case in which a leaf node, such as 4, receives a message for a node with higher address, such as 5, since this situation is impossible: the controller knows that every message for 2, 3, 4 must be routed through 2, every for 5, 6,7 through 5 and ones for 8, 9, 10 through 8.
* Confirmation messages: they are sent by bulbs to controller. If the node isn’t a child of controller, routes the message though the father node; else, it’s routed directly to 1.

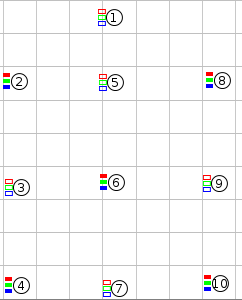
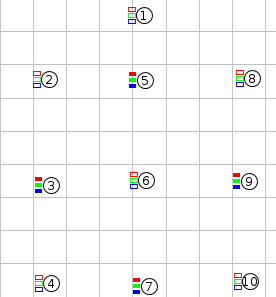
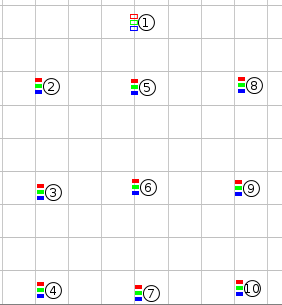
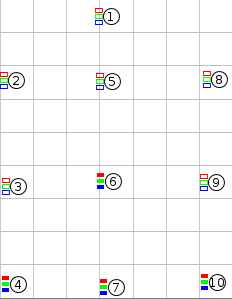
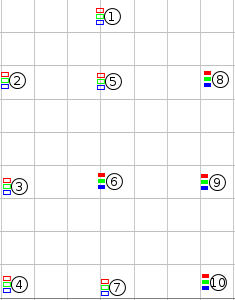
Bulbs Working

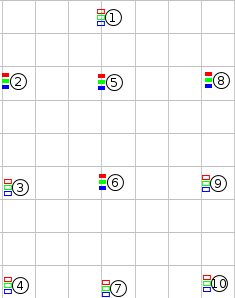
Bulbs cannot do anything without receiving a message. When a message is received, it’s check the type of message:

* Command message. If the message is for the node that received the packet, the led is set to the value contained in the message, and a confirmation is sent to the controller, according to the previously described routing strategy (the addressing is hierarchical, so the controller is surely in a upper node). Otherwise, if the message is for another bulb, it’s redirected using the routing strategy described before.
* Confirmation message: it’s surely a confirmation from another node. It must be redirected to the controlled, using the routing described in the previous paragraph.

Controller Working

Controller implements at the moment three different patterns:

* Cross Switch Pattern
* 
* All Led On Pattern
* 
* Triangular Switch Pattern
* Immagine che contiene testo, verde, colorato, grafica vettoriale

  Descrizione generata automaticamente

For brevity, situations with leds off are omitted, and the sequence can be changed due to the number of maximum iteration. In fact, the variable of the previous pattern is shared between triangle and cross switch, to make the code lighter.

The controller operates as follow:

* A timer is started. Each time the timer is fired, the nextPattern routine is called, and two things happen, since each patterns runs for a fixed number of iteration:
  + If the maximum number of iterations of the same pattern is reached, the pattern is changed, and the routine nextPattern is recalled.
  + Else
    - If all the leds are on, the controller sends a message to all the nodes to turn off the leds
    - Else, calls a routine to start the next pattern.
* When a routine to start the next pattern is called, some operations are done
  + The current node to be processed is reset, and the routine to process next bulb is called
  + If all the bulbs have been processed, the routine ends.
  + Otherwise, the single node is processed, and
    - If the leds must be turn on, a command message is sent, and the controller waits for a confirmation of led processing. Then, the routine for processing next led is called.
    - Otherwise, no message are sent (since all lights are turn off before turning on other leds), and the routine is recalled to process the next led.

In all the message sending case, an ACK is requested, and the message is resent if the ACK is not received.