RGPIO – Remote GPIO java library

This library extends the PI4J library that controls the GPIO pins of a Raspberry PI (RASPI), to include the GPIO pins of remote devices. These remote devices can be ESPxxx or other RASPIs. The remote devices connect to the central RASPI via TCP/IP. The purpose is to easily control a number of distributed physical devices within reach of a WiFi network.

Design principles

* All ESP devices for use with RGPIO are programmed identically, making them interchangeable and easily replaceable.
* Remote devices receive an IP address via DHCP on power-on. They then report to the RASPI and repeat this on regular intervals. So the RasPI knows which devices are alive and can take action if it does not receive updates in time.
* To connect to the RASPI, the remote devices have to know the RASPI’s IP address. (For now it is hard-programmed)

(misschien kan de ESP iets zenden naar het broadcast address 255.255.255.255 dat opgevangen wordt door de RASPI die dan zijn ip adres opstuurt, zie <https://www.arduino.cc/en/Tutorial/WiFiSendReceiveUDPString> packet sturen = de laatste 3 lijnen:

    Udp.beginPacket(Udp.remoteIP(), Udp.remotePort());  
    Udp.write(ReplyBuffer);  
    Udp.endPacket();

)

* Remote devices have a unique hardware identifier from the factory. In a specific environment, they have a certain function – e.g. they control a specific relay switch. We also assign an identifier to this function. The central system needs to be able to couple the hardware identifier to the function identifier. A way to do this is to power up one remote device at a time. RASPI receives the hardware identifier and checks if it is already known. If not, user interaction is required to couple the hardware identifier to a function identifier.
* Function identifiers are assigned by the user of the system to each individual GPIO pin. At power-up, the user will therefore be asked to specify the function of each pin that serves a purpose, for instance, when a 4-pin ESP comes up:

|  |  |  |
| --- | --- | --- |
| 45:22:af:d3 | GPIO 0 | Boiler Relay |
| 45:22:af:d3 | GPIO 1 | Camera on/off |
| 45:22:af:d3 | GPIO 2 |  |
| 45:22:af:d3 | GPIO 3 |  |

* When all remote devices are identified and working, they can be controlled in a uniform way using the function identifier:
  + Pin=RGPIO.getPin(“Camera on/off”)
  + Pin.getState();
  + Pin.setState(Boolean: state);
  + Pin.addListener(RGPIOlistener());
  + … analoge pinnen ?