Background:
More and more IoT devices are implemented in modern household. Smart light control system, air conditioners, curtains, etc.

Method:
Information from sensors are collected and sent to network devices, network devices process the information and then send command to light, ac, curtains…

Also contain interface for control order from user

Sensor information can be time,temp,humidity… eg 1600,23,56……

control command can be ac t20 h50, light1 on, curtain2 close…

after receiving a user command, the analyze of information from the related sensor can be suspend for a certain length of time, this time can be user defined and having a default value.

For now we don't have any physical device we can use command line for sensor input and user input and observe status output from command line.
Two python program can be used, one constantly generating random environment or sensor information, the other sending user command.

Packet sent from both python terminals should be
time info_type(sensor: temp, hum; user: light_ctl, ac_ctl, curtain_ctl) value temp, hum, light_status, ac_status, curtain_status

logic:
light turn on after 1800, turn off after 2300
curtain close after 2300, open after 0800

ac set to cooling when temp > 25, set to heating when temp < 14, stop working in between
set to humidifying when hum < 30, set to dehumidifying when hum > 70, stop in between.

The above is what I originally thought what I can do, but I have only completed the light part.

Implementation:

Header: ethernet, ihome
In ihome header, there are p, 4, ver, from the original calc.p4, and cmd for command type, time to carry time information from sensor/lab computer, status to pass the status back to lab computer or to control a actual light.

A register is declared to store the current light status globally at pi.

There are 4 types of command:
'U' for update. Pi determine next light status based on time information passed to it.
    Logic: switch on when 1800<time<2300
    'U' command is sent by the clock program at fixed frequency and with time informaiton.
'1' and '0' for user on and off. When user send '1' or '0' the light will be turn on or off for a customized length of time.

‘C’ for check. User can check the current light status by sending check commend.

There are 4 types of status:
‘0’ for off
‘1’ for on
‘2’ and ‘3’ for user_off and user_on, when light is switched off or on by user, status change to user_off or user_on, and when user status is received by clock program, it will stop sending update command for one cycle.

The python program iot_home_clock generate time information and send to Piand display real time light status based on status sent back from pi
The python program iot_home is used to send user command, which can be ‘1’,’0’ or ‘C’