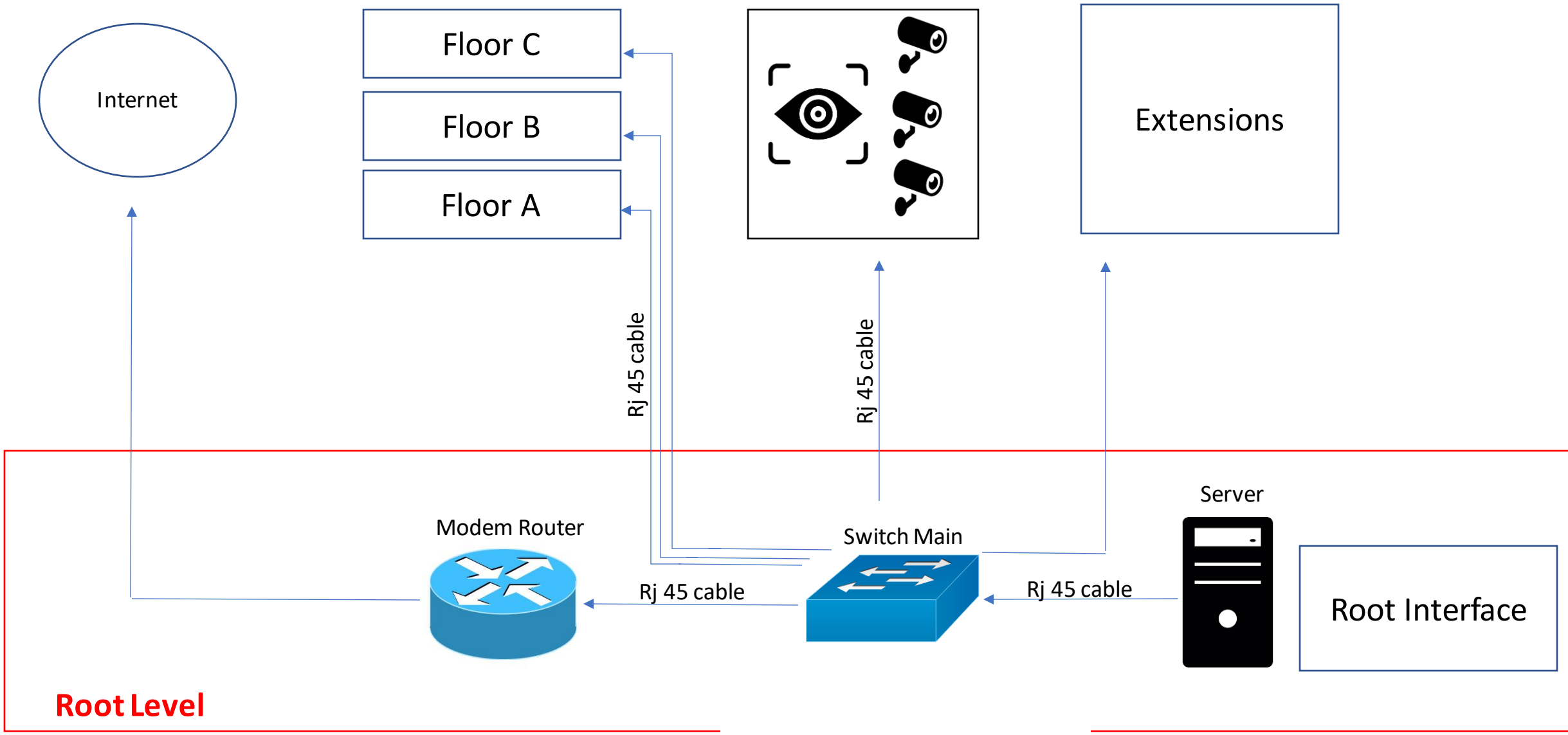
System walkthrough

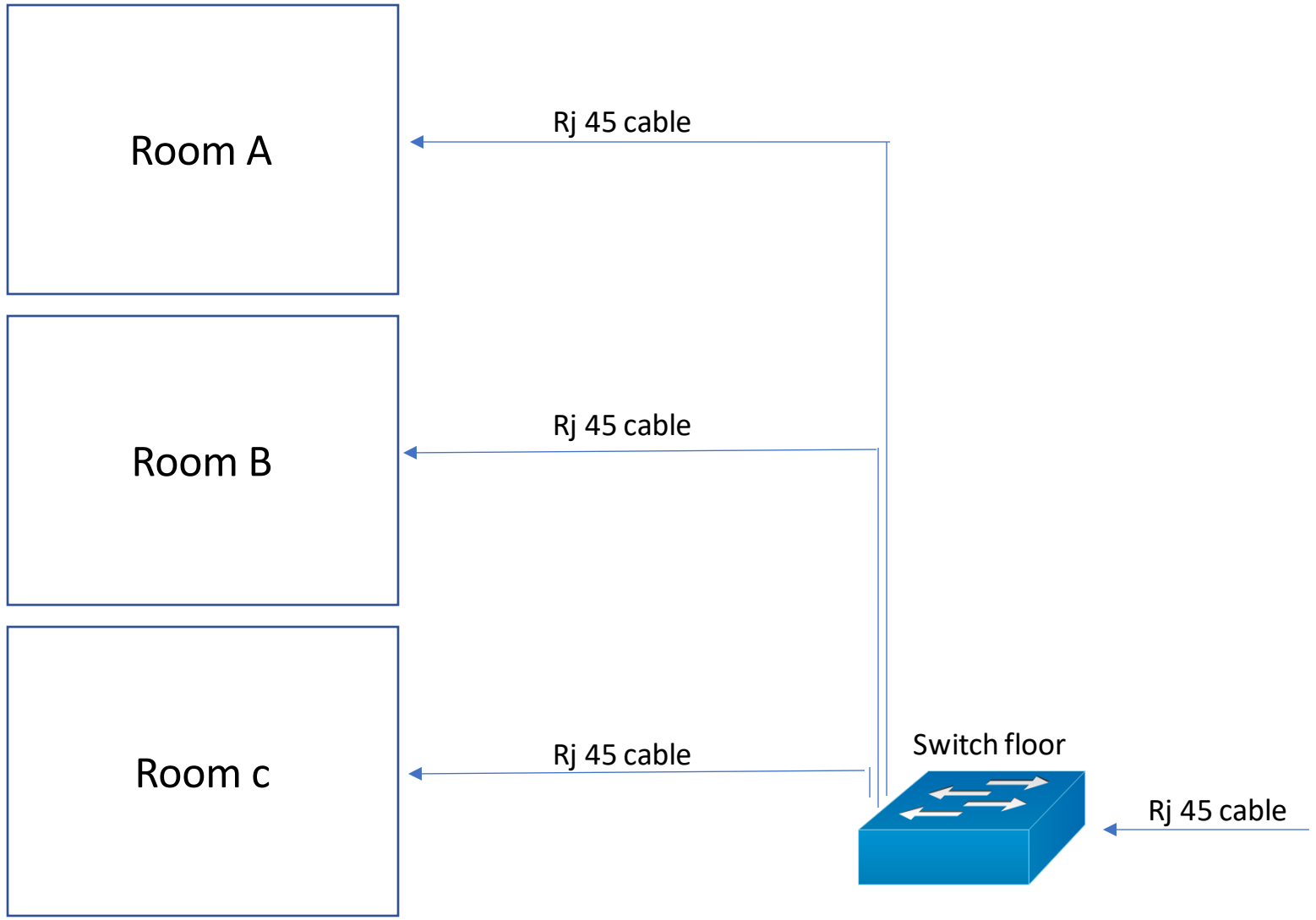
Design

Under this title a detailed schemes and plans are projected to explain the chosen design for our system

General layout



Floors Layout



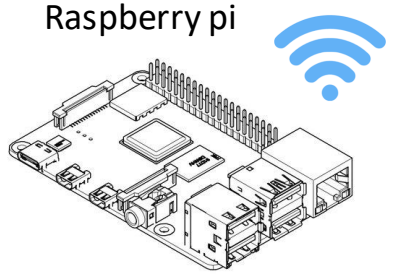
Building

Room layout

Room x

Wireless connection

Raspberry pi



Guest Control

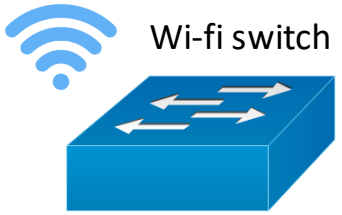
Electrical wiring

Adaptation circuit

Electrical wiring

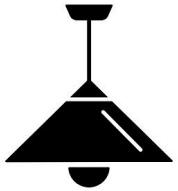
Root & Classic control

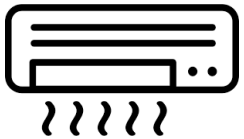
Wireless connection

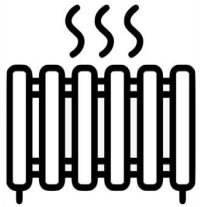


Wi-fi switch

Rj 45 cable

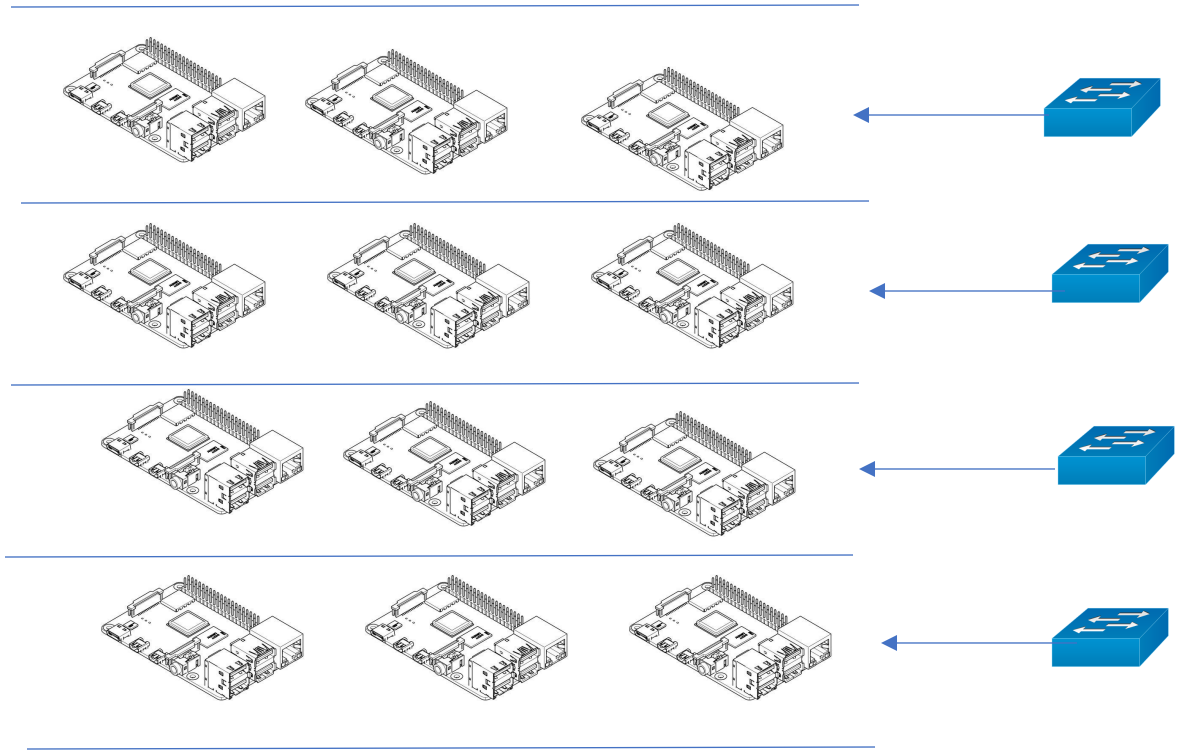


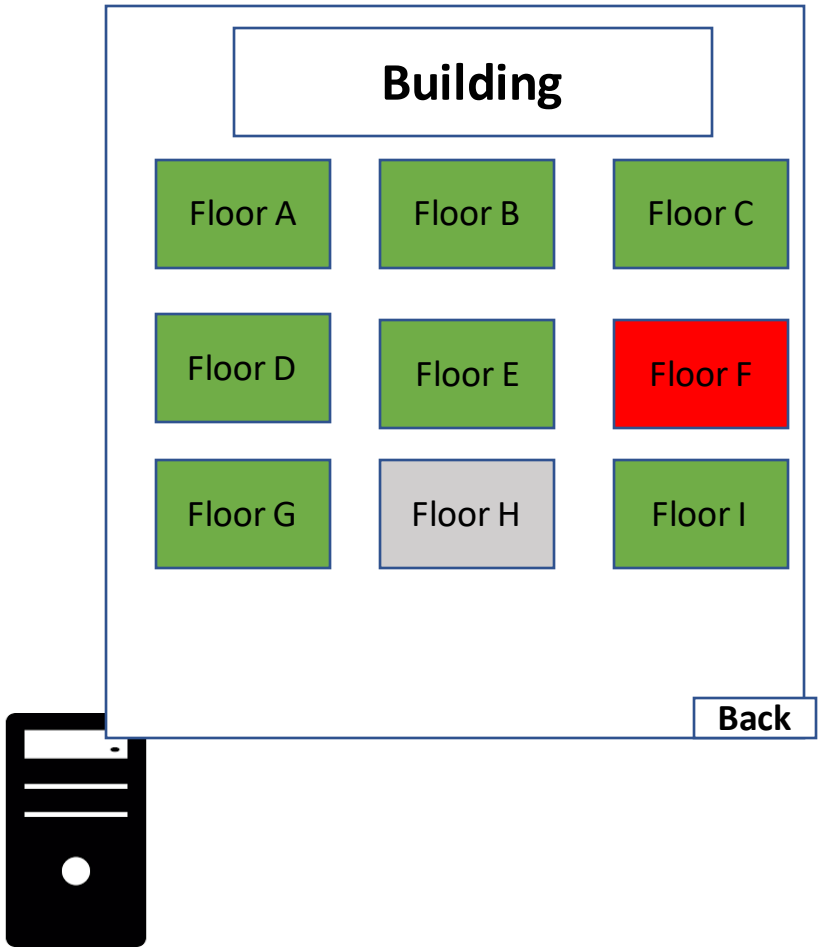




Room Out/Input

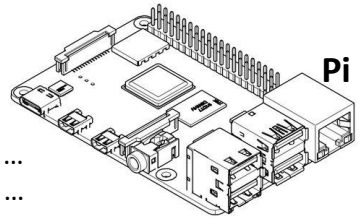
Rooms control Root.interface





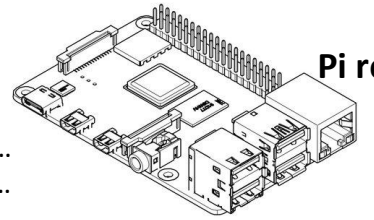
Rooms control Root.interface

Ip adress: 192.168.x.x
Hostname: PiFloorBroomA
Nfc UID: xx xx xx xx xx xx xx xx
Logfile: PiFloorBroomAlog.txt
Outputs: outp1, outp2, outp3, …
Inputs: input1, input2, input3, …
.
.
.



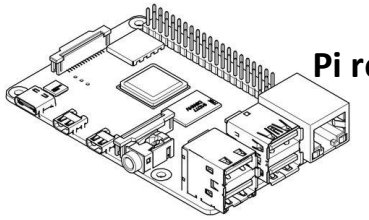
Pi room A

Ip adress: 192.168.x.x
Hostname: PiFloorBroomB
Nfc UID: xx xx xx xx xx xx xx xx
Logfile: PiFloorCroomClog.txt
Outputs: outp1, outp2, outp3, …
Inputs: input1, input2, input3, …
.
.
.

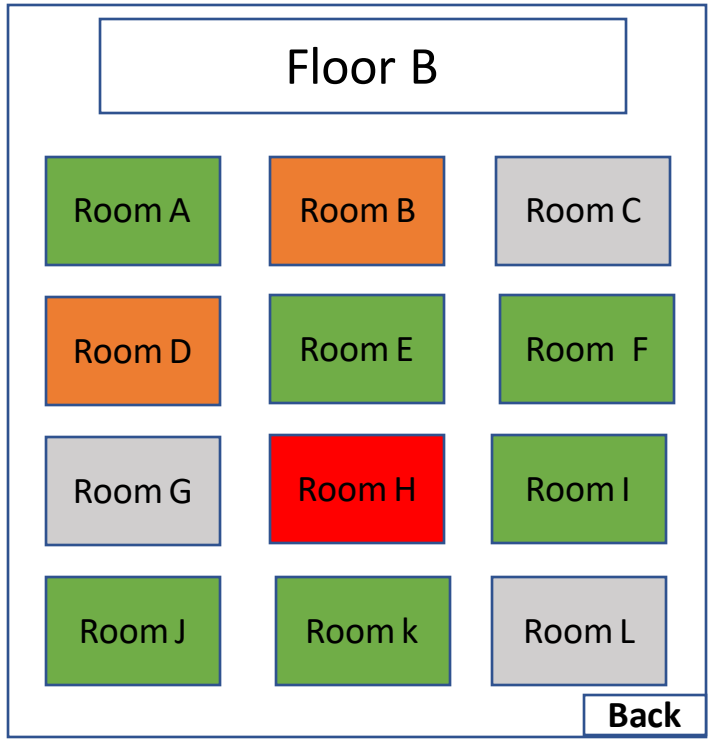


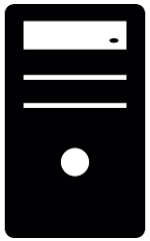
Pi room B

Ip adress: 192.168.x.x
Hostname: PiFloorBroomC
Nfc UID: xx xx xx xx xx xx xx xx
Logfile: PiFloorCroomClog.txt
Outputs: outp1, outp2, outp3, …
Inputs: input1, input2, input3, …
.
.
.



Pi room C





My room

ON

OFF

Other
Outputs/inputs

ON

55°C

OFF

Back

Guest Control

Floor B, Room A

Host interface

Root Control

Wireless
Connection

Floor B, Room A

ON

Internet ON

OFF

Room Control ON

Other
Outputs/inputs

ON

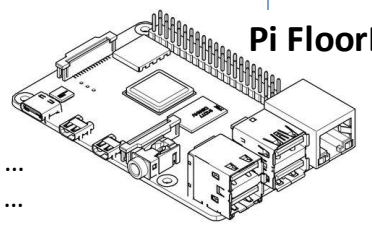
55°C

OFF

Back

Root interface

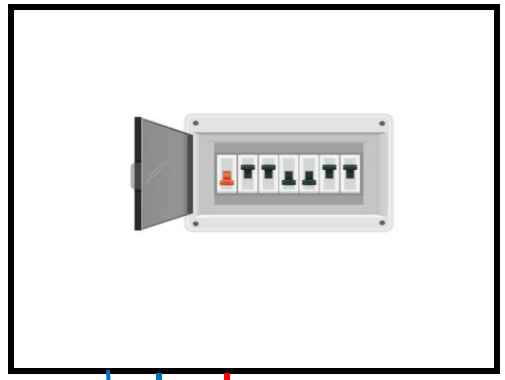
Ip adress: 192.168.x.x
Hostname: PiFloorBroomA
Nfc UID: xxxxxxxx
Logfile: PiFloorBroomAlog.txt
Outputs: outp1, outp2, outp3, ...
Inputs: input1, input2, input3, ...
.
.
.



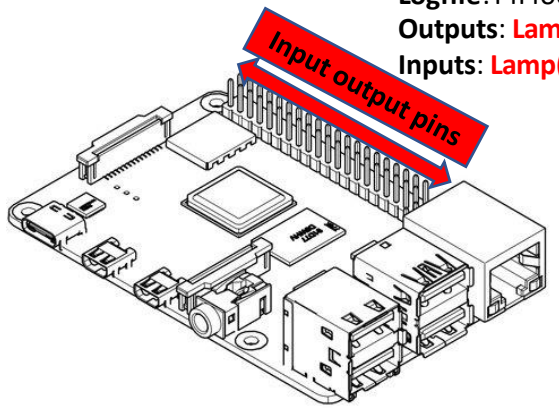
Pi FloorBroom A

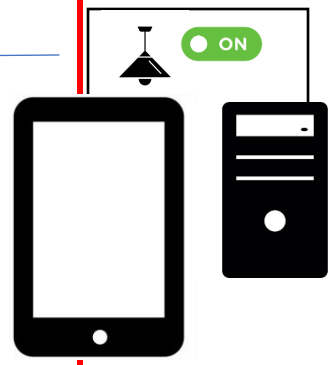
Cable
Connection

Floor B, Room A

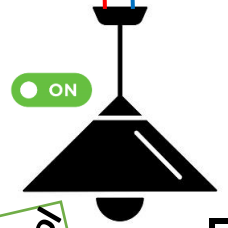


Ip adress: 192.168.x.x
Hostname: PiFloorBroomA
Nfc UID: xx xx xx xx xx xx xx xx
Logfile: PiFloorBroomAlog.txt
Outputs: Lamp(GPIO26 = out.HIGH)
Inputs: Lamp(GPIO04 = in.True)



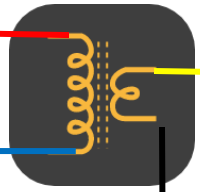


Root & Guest Controll



Classic Control

Adaptation circuit



Transformer Ac 220v to Dc 3v

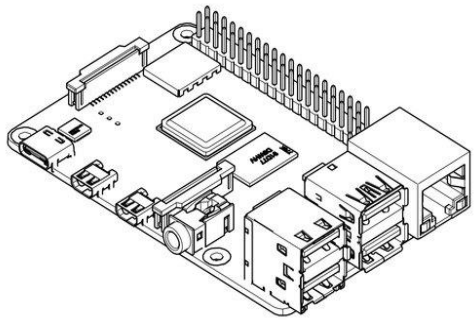
Relay

Resistor

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21

Input output pins

Ip adress: 192.168.x.x
Hostname: PiFloorBroomA
Nfc UID: 00020001 (Decimal-26 bits)
Logfile: PiFloorBroomAlog.txt
Outputs: outp1, outp2, outp3, ...
Inputs: input1, input2, input3, ...



PiFloorBroomA

Running Thread

```
rms.FloorB_RoomA_ID_local = "00020001"
def Lock(self, dt):
    if rms.lockID_read == rms.FloorB_RoomA_ID_local
        open()
```

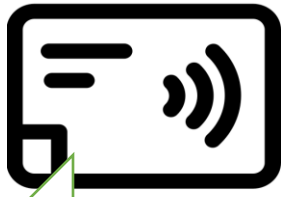
RFID Module

7 Wires connection

NFC

Rms.lockID_Read

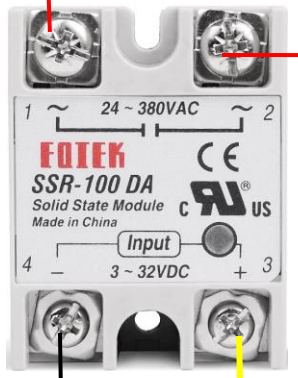
Nfc card ID : 00020001

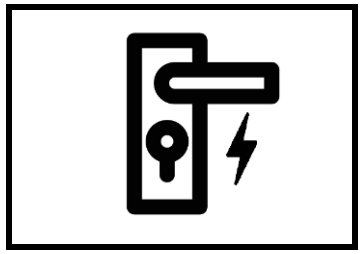


Guest Control

Input output pins

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21



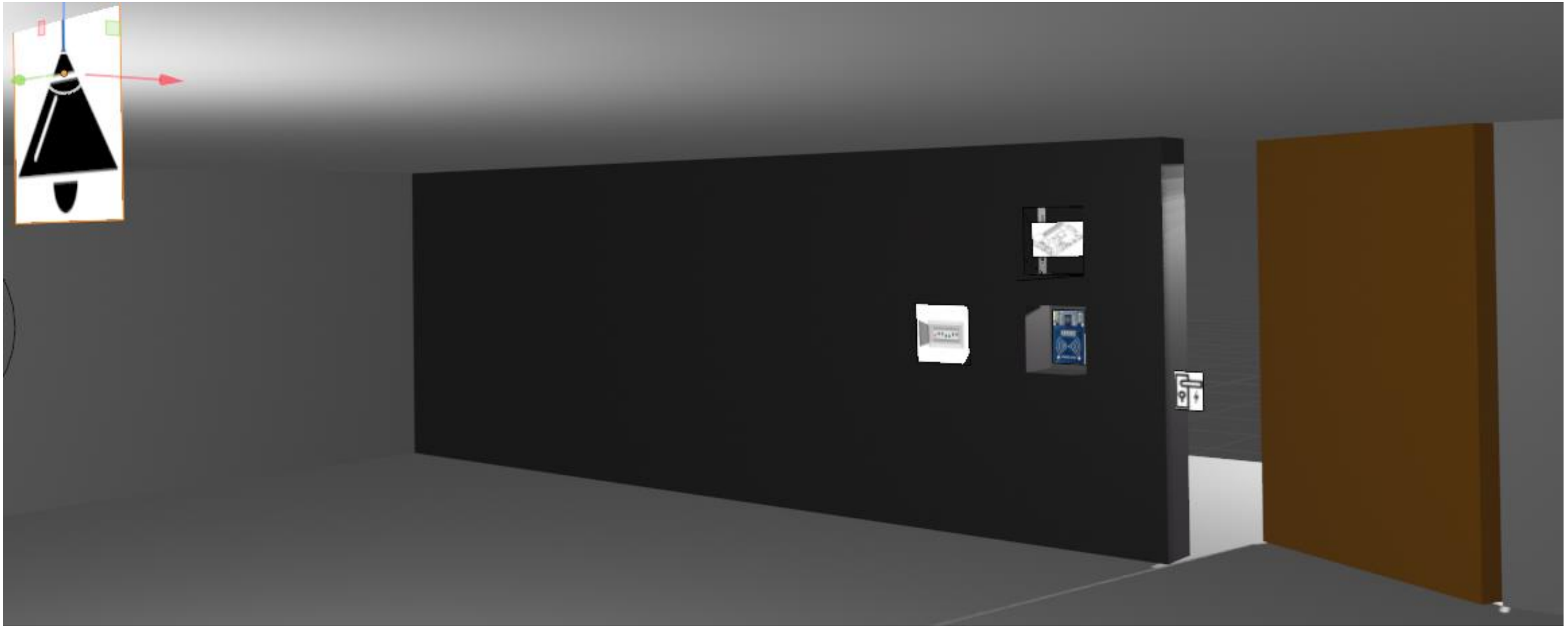


N

F

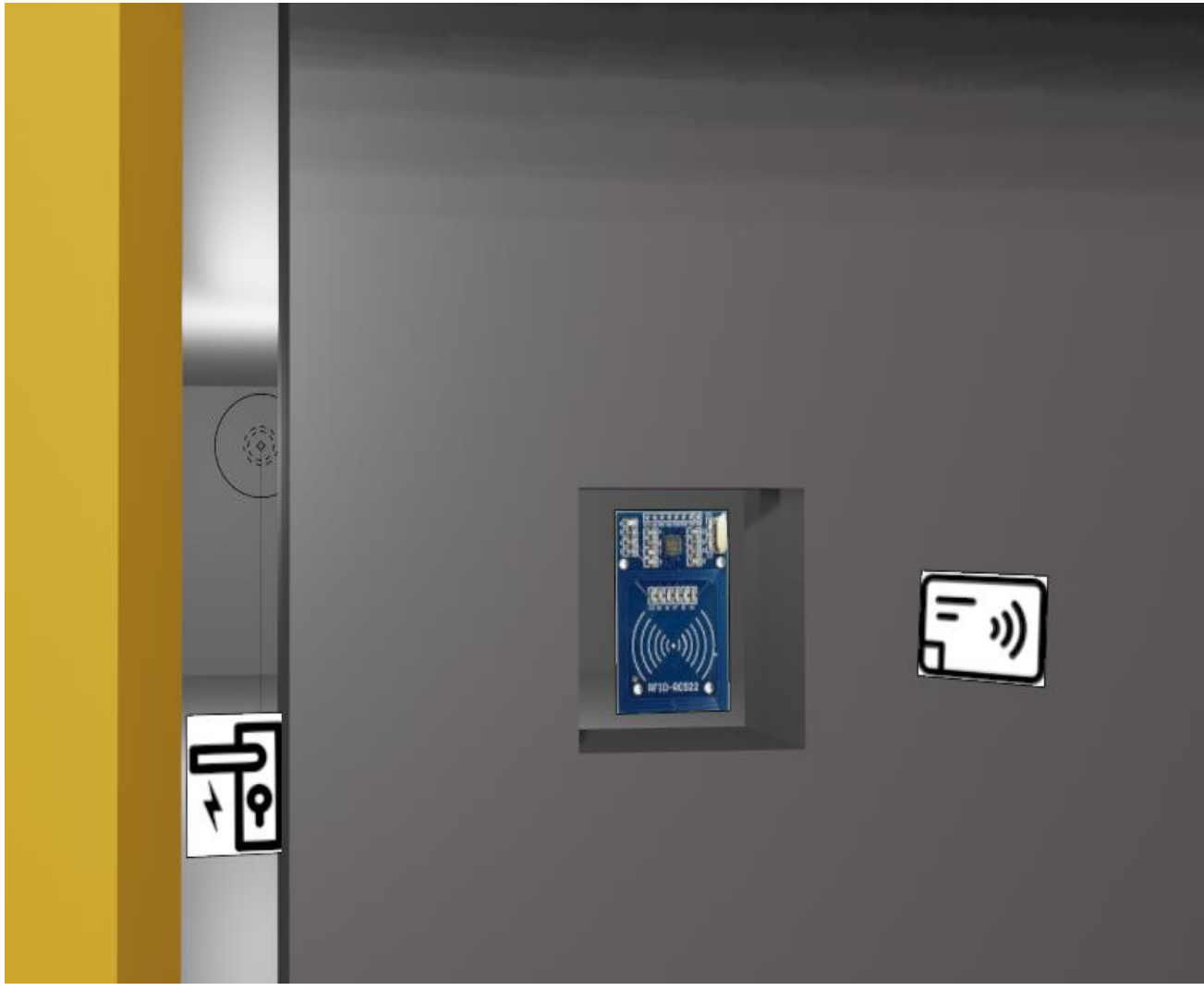
`open()`

Physical example

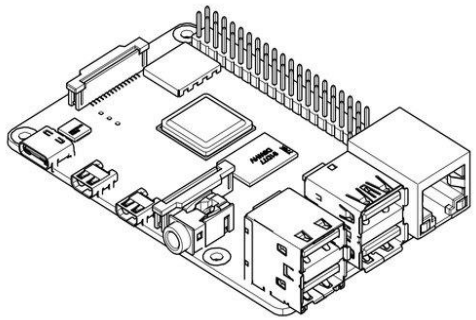


INSIDE example

Outside example



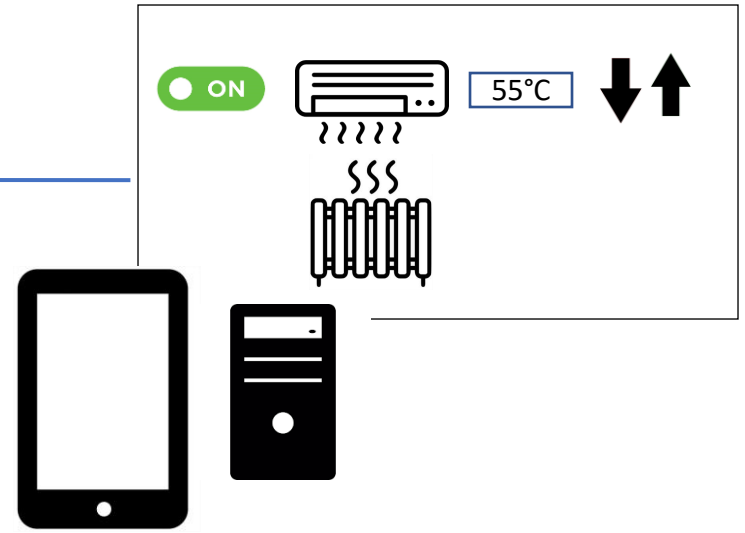
Ip adress: 192.168.x.x
Hostname: PiFloorBroomA
Nfc UID: **00020001 (Decimal-26 bits)**
Logfile: PiFloorBroomAlog.txt
Outputs: outp1, outp2, outp3, ...
Inputs: input1, input2, input3, ...



PiFloorBroomA

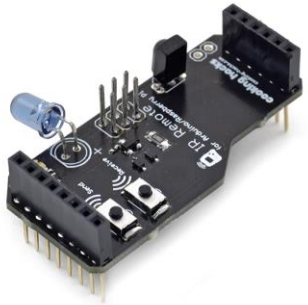
Running Thread

```
def HVAC_remote(self, dt):
    if self.action == "on":
        rms.FloorB_RoomA_transmit = "On"
    elif self.action == "plus":
        rms.FloorB_RoomA_transmit = "plus"
    elif self. #.........
```

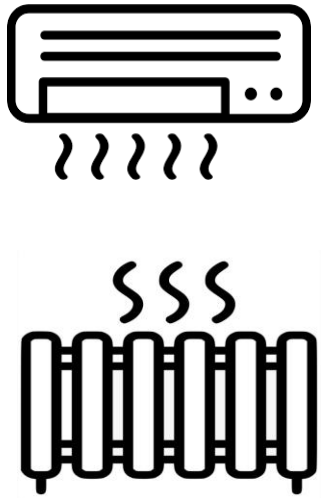


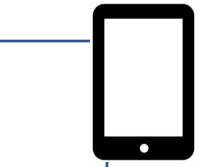
Input output pins

	Pin No.		
3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21



HVAC IR Remote module





Software Side

Os: android
Program: Room service Guest.**app**
Log: Tab_FloorxRoomxLog.**txt**

Hardware:
Wi-fi: capable
Nfc: capable

Guest Control

Root Control

Software Side

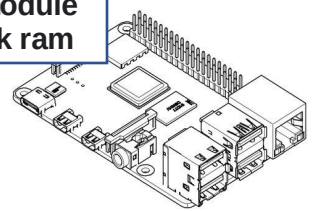
Os: Debian
Program: RMSt.**py,** Surviellance.**py,** Extenstions.py
Log: Rms_Log.**txt**

Hardware
Nfc: capable
Ethernet: Multiple cards
Cpu unit: Supper cpu multiple Cores
Gpu: Multiple Gpu units
Display: Multiple high resolution screens
Memory: High Ram and Rom with externel drives

Software side

Os: Raspbian
Program: PI_RMS.**py**
Log: Pi_FloorxRoomxLog.**txt**

Hardware:
Ethernet: capable
Wi-fi: capable
Modules: nfc module, Havc module
Memory: 16 gb Rom and stock ram



Software Side

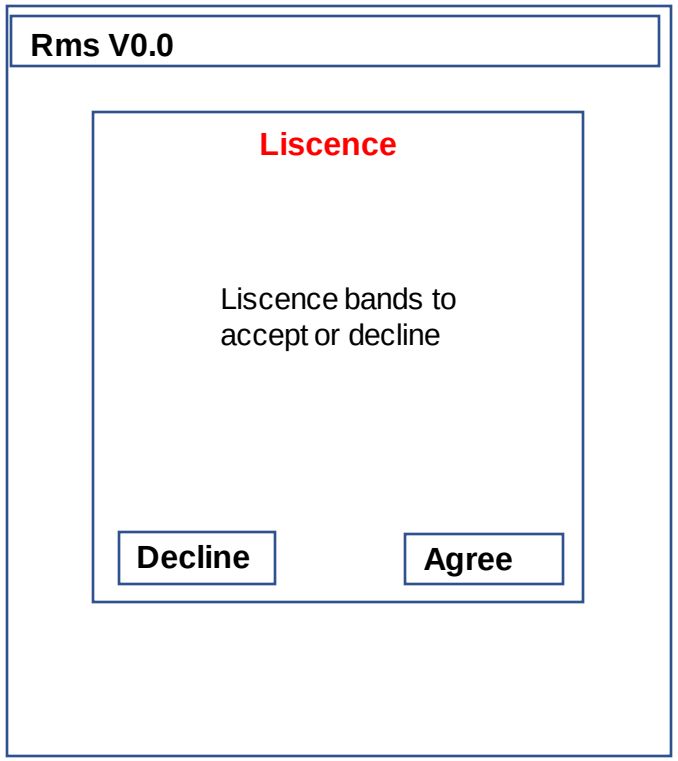
Os: Gnu Debian

Program: RMS.py (Packages: Python, Kivy, Sockets, encryption , ssh, apache)

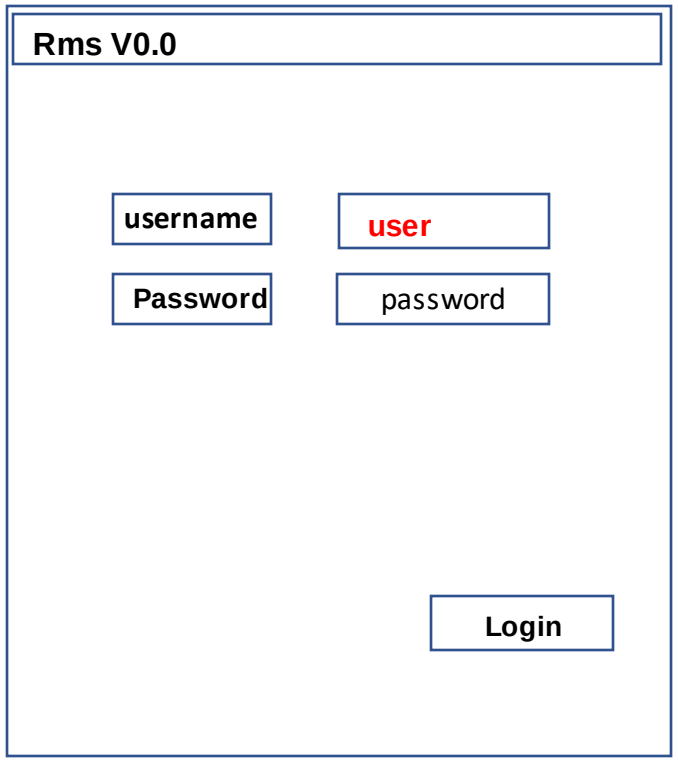
Debian **User:CompanyX**

Execute First time

Execute



Agree

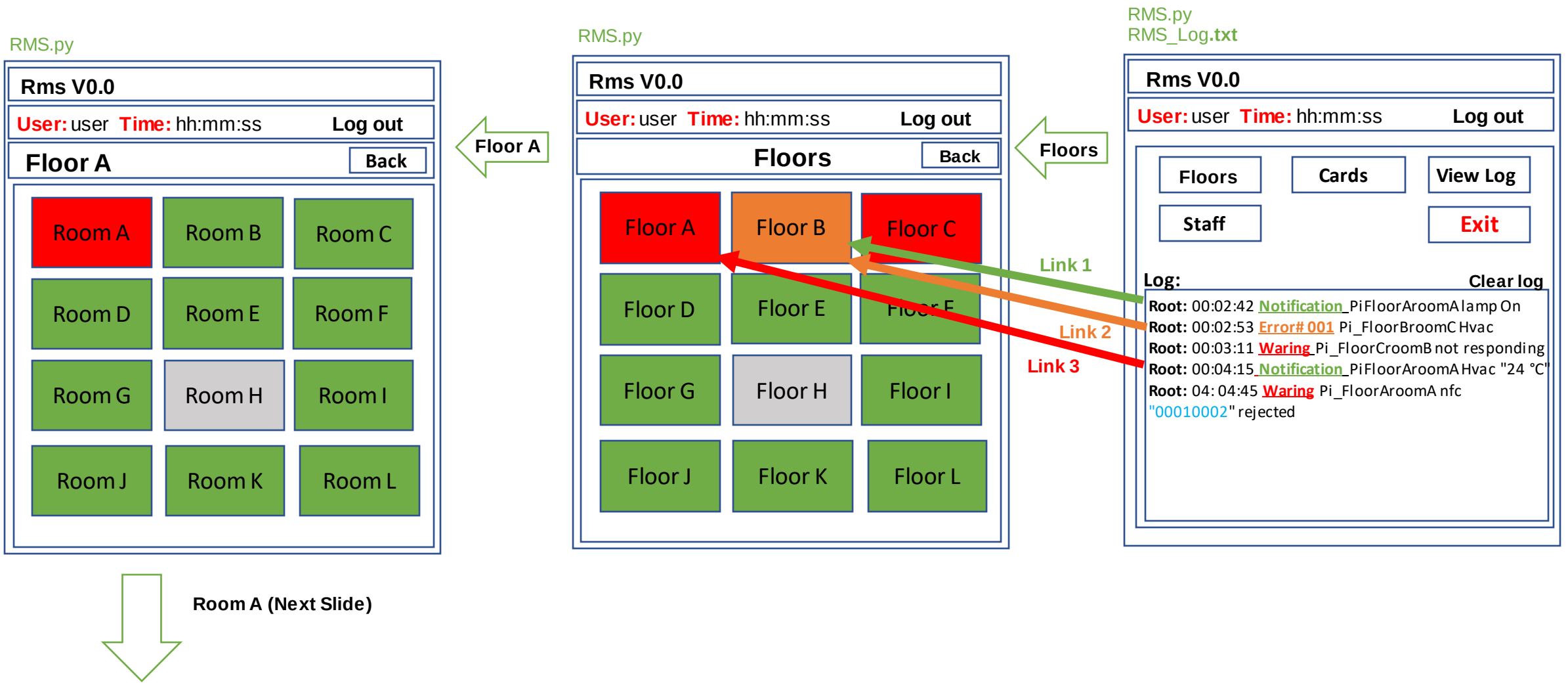


Login (Next slide)

user

Note: The Interface can customizable on request and it is environement dependable

Root RMS.py interface



Note: Log prompt Generates three level **link** messages notification errors and Warinings

Root RMS.py interface

RMS_Log.**txt**

LOG

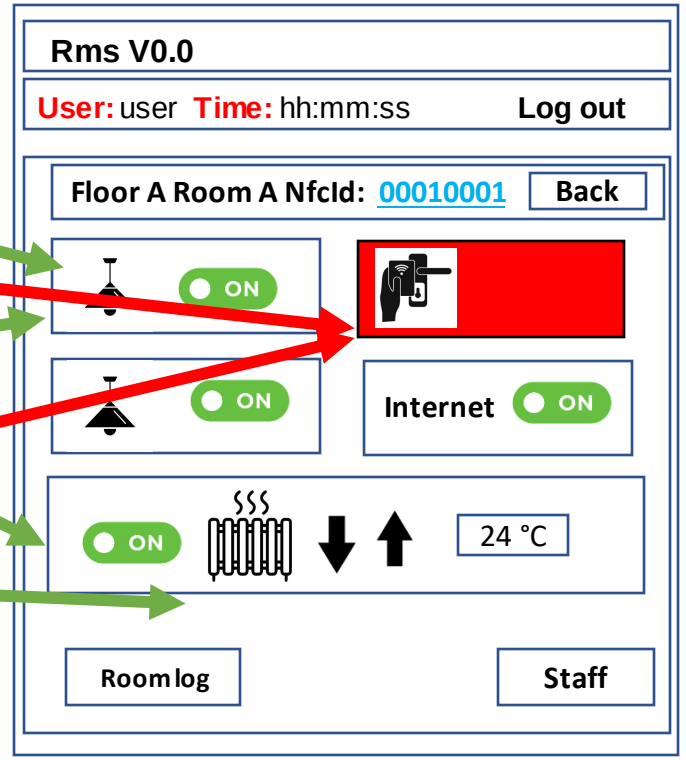
Root: 00:02:42 **Notification** PiFloorAroomA lamp On
Root: 00:02:53 **Error# 001** Pi_FloorBroomC Hvac
Root: 00:03:11 **Waring** Pi_FloorCroomB not responding
Root: 00:04:15 **Notification** PiFloorAroomA Hvac "24 °C"
Root: 04: 04:45 **Waring** Pi_FloorAroomA nfc "00010002" rejected

Pi_FloorxRoomxLog.**txt**

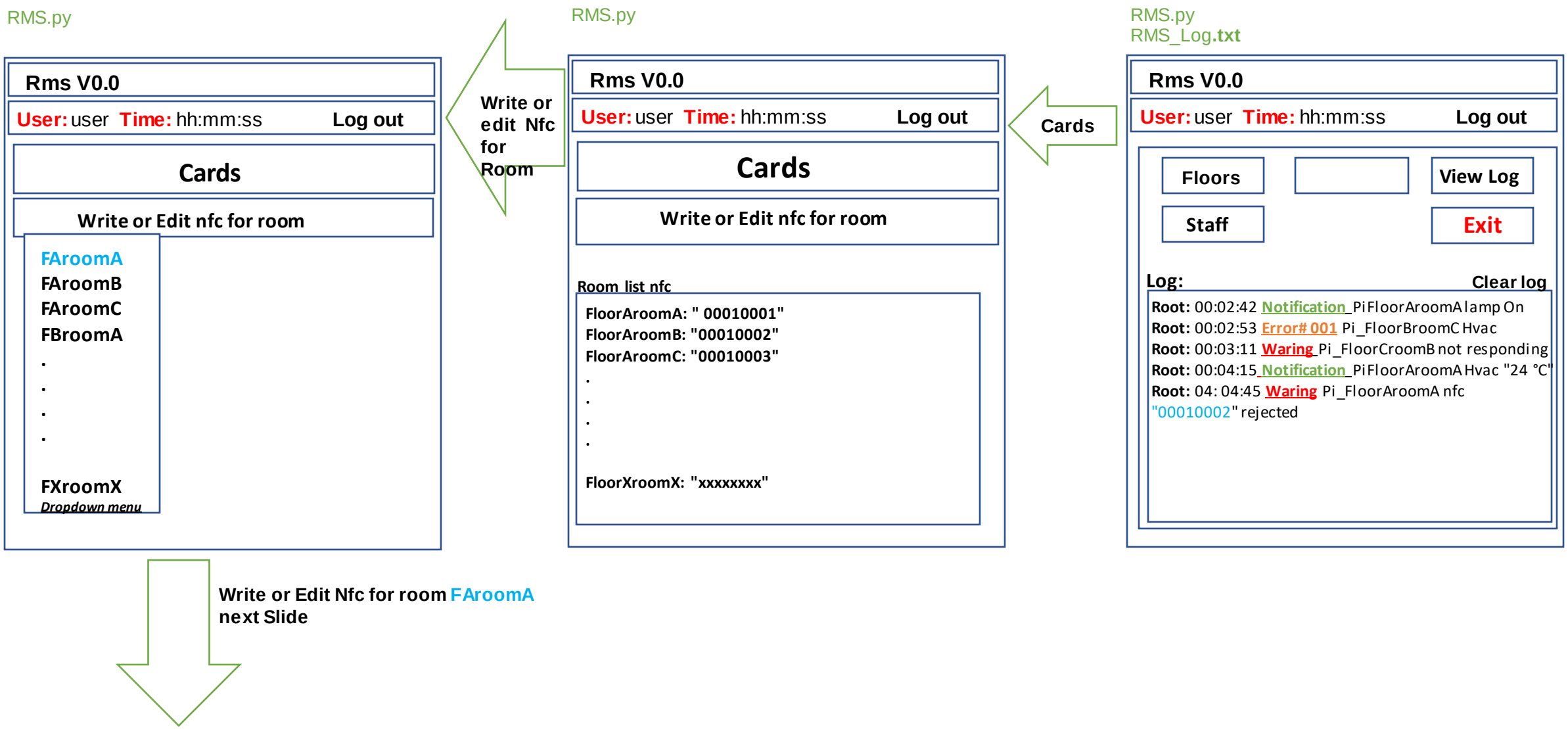
Room log

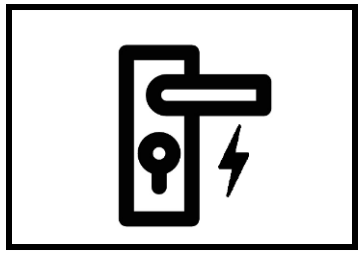
PiFoorAroomA: 00:2:41 **Notification** **Lamp(GPIO26 = out.HIGH) Lamp(GPIO04 = in.True)**
PiFoorAroomA: 00:04:14 **Notification** **HVAC_Remote Plus "24 °C"**
PiFoorAroomA: 04:04:44 **Waring** **nfc** **"00010002"** **rejected**

RMS.py
Pi_FloorxRoomxLog.**txt**

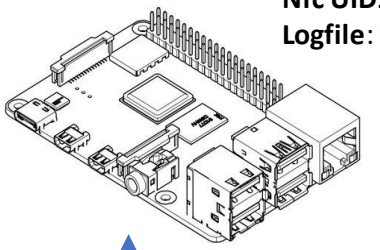


Note: nfc refrence is a **link** to cards


RMS.py
Rms V0.0
User: user Time: hh:mm:ss
Log out
Cards
Write or Edit nfc for room
FAroomA
FAroomB
FAroomC
FBroomA
.
.
.
.
FXroomX
Dropdown menu
Write or edit Nfc for Room
RMS.py
Rms V0.0
User: user Time: hh:mm:ss
Log out
Cards
Write or Edit nfc for room
Room list nfc
FloorAroomA: " 00010001"
FloorAroomB: "00010002"
FloorAroomC: "00010003"
.
.
.
.
FloorXroomX: "xxxxxxxx"
Cards
RMS.py
RMS_Log.txt
Rms V0.0
User: user Time: hh:mm:ss
Log out
Floors
View Log
Staff
Exit
Log:
Clear log
Root: 00:02:42 Notification_PiFloorAroomA lamp On
Root: 00:02:53 Error# 001 Pi_FloorBroomC Hvac
Root: 00:03:11 Waring_Pi_FloorCroomB not responding
Root: 00:04:15_Notification_PiFloorAroomA Hvac "24 °C"
Root: 04: 04:45 Waring Pi_FloorAroomA nfc
"00010002" rejected
Write or Edit Nfc for room FAroomA
next Slide



open()



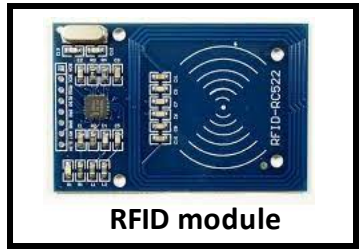
Ip adress: 192.168.x.x
Hostname: PiFloorAroomA
Nfc UID: **00020001** Swapped to **New Value**
Logfile: PiFloorBroomAlog.txt

update

7 Wires connection

Nfc card ID :

Edited value or written value



RFID module

NFC

Rms.lockID_Read

update

RMS.py

Rms V0.0

User: user **Time:** hh:mm:ss **Log out**

Cards **Write or Edit Nfc for FAroomA**

Enter a decimal value **Encode to card**

Place your card near to nfc reader

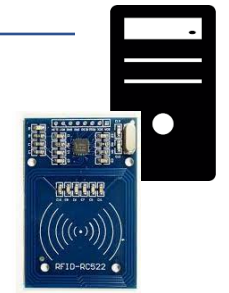
Scan card

Place your card near to nfc reader

Old nfc Id: 00020001 **Update** **Back**

update

Update next Slide

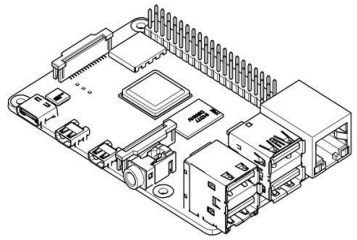


RFID module

Note: Each PiFloorXroomX has its own nfc UID stored in it to be handeled with **PI_RMS.PY**

Ip adress: 192.168.x.x
Hostname: PiFloorAroomA
Nfc UID: New Value

Logfile: PiFloorBroomAlog.txt



Os: Android
Program: Room service Guest.app

Log In

Log In

Room service Guest

ON

ON

Internet ON

ON 24 °C

Room service Guest

Scan Card

ID: New Value

Log In

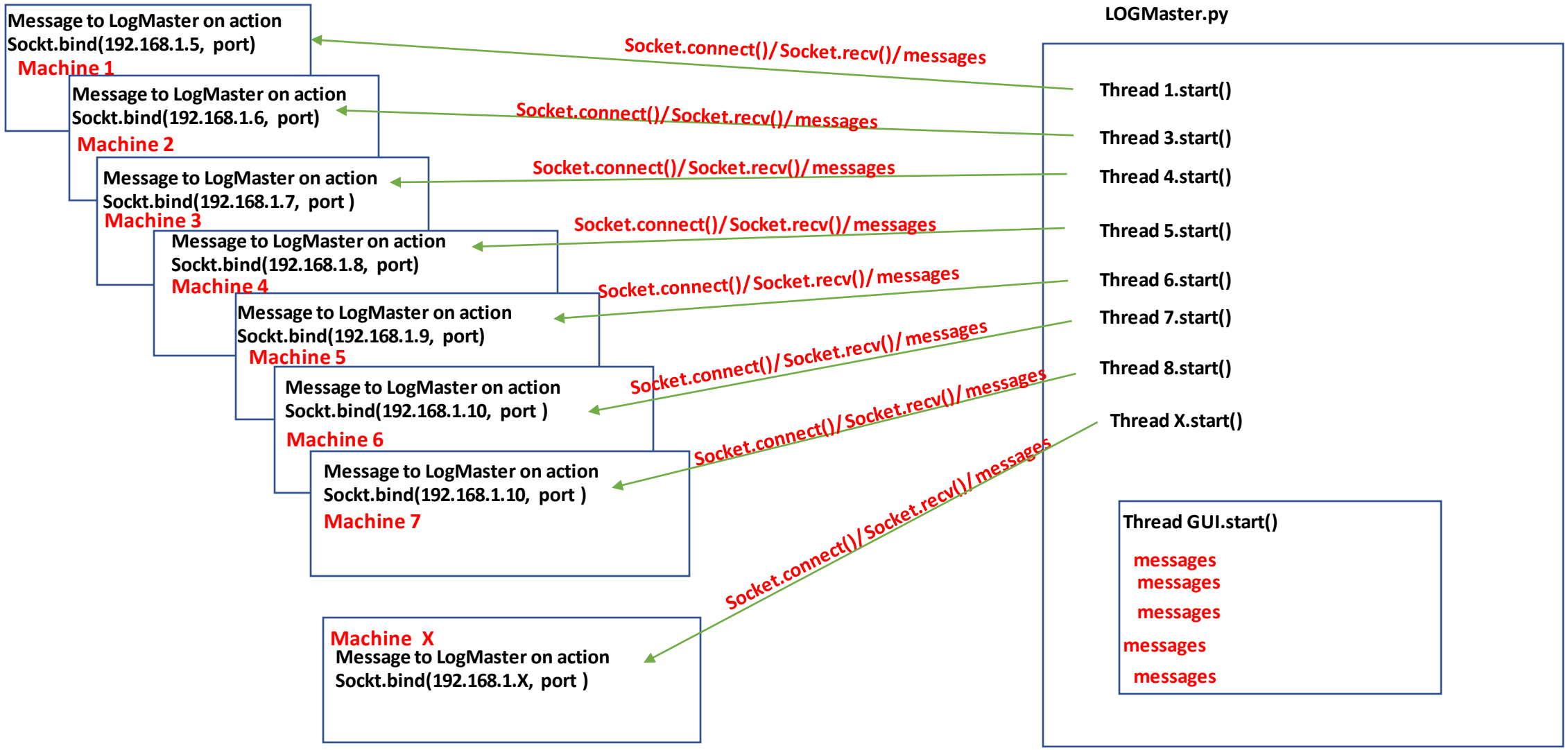
Cant login

Place your card or phone near to Tablet

Nfc card ID : New Value

NFC

Read Card


Message to LogMaster on action
Sockt.bind(192.168.1.5, port)
Machine 1
Message to LogMaster on action
Sockt.bind(192.168.1.6, port)
Machine 2
Message to LogMaster on action
Sockt.bind(192.168.1.7, port)
Machine 3
Message to LogMaster on action
Sockt.bind(192.168.1.8, port)
Machine 4
Message to LogMaster on action
Sockt.bind(192.168.1.9, port)
Machine 5
Message to LogMaster on action
Sockt.bind(192.168.1.10, port)
Machine 6
Message to LogMaster on action
Sockt.bind(192.168.1.10, port)
Machine 7
Machine X
Message to LogMaster on action
Sockt.bind(192.168.1.X, port)
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
Socket.connect()/Socket.recv()/messages
LOGMaster.py
Thread 1.start()
Thread 3.start()
Thread 4.start()
Thread 5.start()
Thread 6.start()
Thread 7.start()
Thread 8.start()
Thread X.start()
Thread GUI.start()
messages
messages
messages
messages
messages

Linux oriented system Table on three rooms per floor, 3 floors

Parts	quantity	Unit price	Quantity price
Adaptation circuits	9 pcs		
Relays	36 pcs		
Solid state relays	27 pcs		
Nfc modules	9 pcs		
Heater modules	9 pcs		
Air con modules	9 pcs		
Sensors	27 pcs		
Electrical Cables			
Tablets	9 pcs		
Wireless switches	9 pcs		
Swiches	4 pcs		
Rj 45 cables			
Electrical locks	9 pcs		
Softwares	3 pcs		
Software extensions			
Servers	1 pc		
Raspberry pis	9 pcs		
Nfc Cards			

Parts	quantity	Unit price	Quantity price
Ac 220V to Dc 3V	27 pcs		
Working Hours			

Total Cost	