

**Echipa 4:**

Ciorățanu Maria  
Pâncă Aida-Gabriela  
Varzar Alina-Miruna

**Descriere proiect:**

Smart School este o școală modernă care integrează tehnologie inteligentă pentru a face învățarea mai interactivă și eficientă. Proiectul include table interactive, ochelari VR, laptopuri și tablete pentru elevi, sisteme audio pentru profesori, securitate inteligentă cu camere și acces biometric, precum și platforme digitale de învățare. Toate acestea ajută la crearea unui mediu educațional conectat, unde lecțiile devin mai atractive, iar elevii pot învăța într-un mod mai intuitiv și interactiv.

 **Observație!** În cadrul acestei documentații, pentru început am prezentat toate obiectele pe care le-am încorporat în proiectul nostru. Pentru fiecare obiect am inclus denumirea, imaginea (din simulator) și funcționalitățile sale. Mai apoi, la finalul documentului, am prezentat partea de conectivitate, care include:

- ce tip de conexiune am ales
- ce alte configurații au fost necesare
- ce am reușit să realizăm pe partea de programabilitate (partea de cod)
- pentru fiecare acțiune la care am utilizat tutoriale video, am trecut note de subsol cu link-urile respective.

Am dori să spunem că, deoarece proiectul nostru este destul de complex, cu multe obiecte care sunt interconectate, conectate la internet, care pot comunica unele cu altele și care au condiționări, nu am reușit să scriem foarte detaliat toate conexiunile la care am lucrat, deoarece au fost foarte mulți pași pe care îi repetam, încercam de mai multe ori sau revineam la ei. Cu toate acestea, ne-am străduit să explicăm cât de bine am putut ce am făcut, am pus gif-uri, videoclipuri, poze și am trimis proiectul pe formularul din Laboratorul 4.

În legătură cu contribuția fiecăreia dintre noi la proiect, am muncit în mod egal, ne-am întâlnit și am lucrat împreună la proiect de fiecare dată, ne-am sfătuit și nu am putea spune cine a făcut fiecare funcționalitate, deoarece a fost un proces mai lung, am combinat tutoriale pe care le-am făcut pe parcurs și am încercat să îmbunătățim cât de mult am putut.

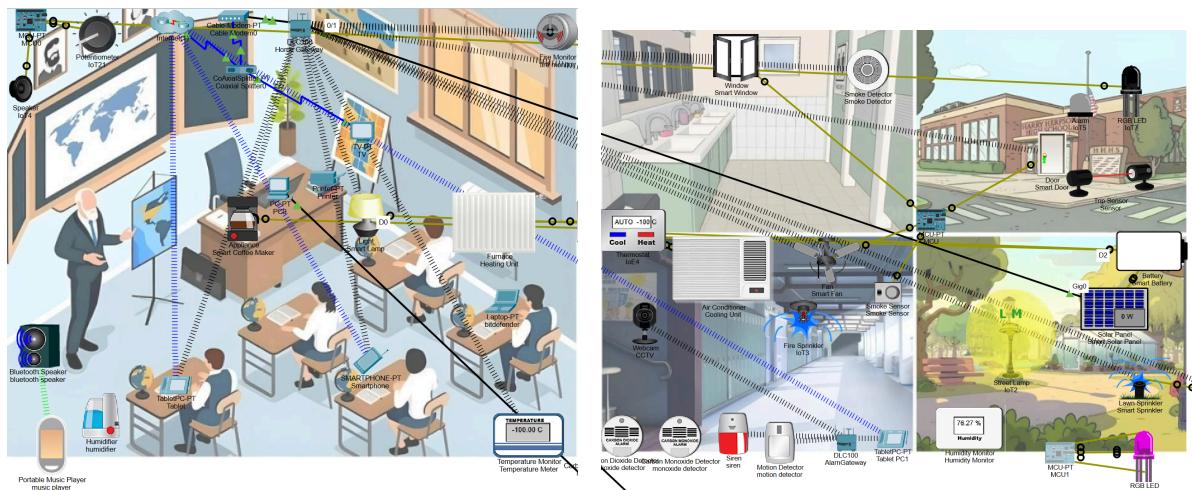
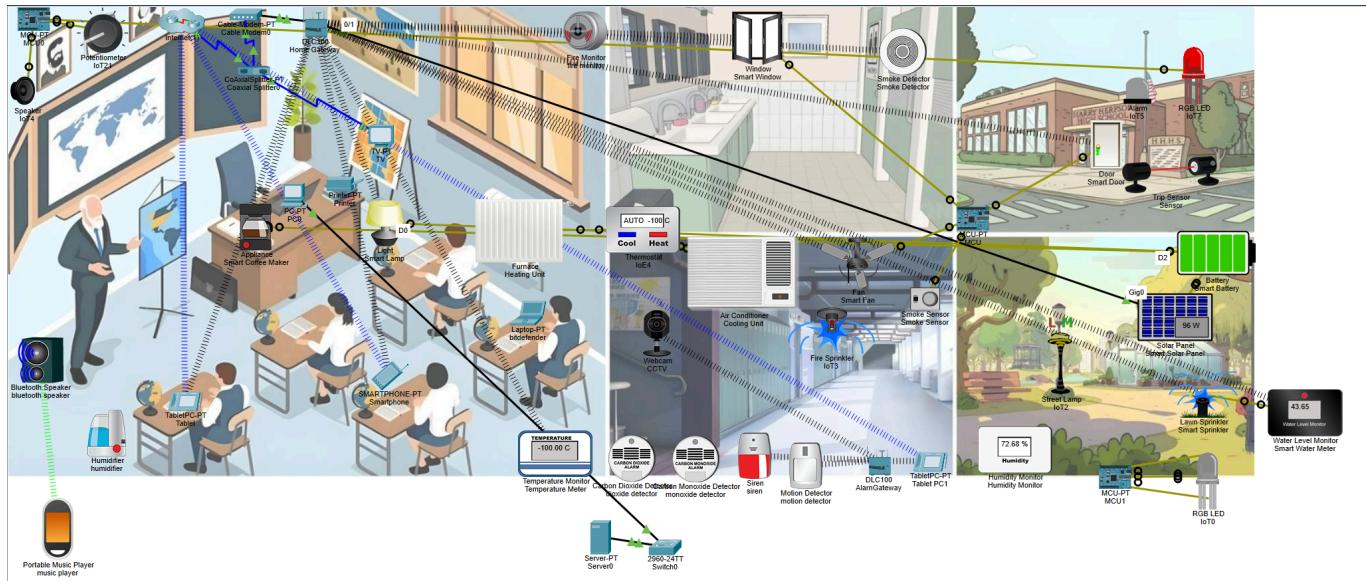
**Înțelegerea dispozitivelor care compun școala intelligentă**

Home Gateway acționează ca un concentrator și router pentru toate dispozitivele interne. De asemenea, oferă o interfață web care permite utilizatorilor să monitorizeze și să controleze diverse dispozitive inteligente din casă. Dispozitivele inteligente se pot conecta la Home Gateway prin conexiuni wireless și/sau prin cablu.

Dispozitivele din casa inteligentă pot fi monitorizate și controlate de la distanță prin orice calculator din casă. Deoarece toate dispozitivele inteligente se conectează la Home Gateway, care găzduiește o interfață web, tabletele, telefoanele, laptopurile, calculatoarele desktop sunt folosite pentru a interacționa cu ele.

## Cum facem conexiunea

1. Dă clic pe Tabletă. (Tableta este așezată pe o bancă în sala de clasă.)
2. Navighează la: **Desktop > Web Browser**
3. În bara de adrese, scrie **192.168.25.1** și apasă **Enter**. Aceasta este adresa IP a Home Gateway.
4. Folosește **admin/admin** ca nume de utilizator și parolă pentru a te autentifica în Home Gateway.



## 1. Boxă

The screenshot shows the IoT6 software interface with the "Home Speaker" component selected. The window has tabs for "Specifications" and "Attributes". The "Specifications" tab is active, displaying the following details:

- Home Speaker**
- Play sounds.
- Features:**
  - Analog input from 0 to 255 converted to predefined sounds.
  - Sets a SOUND\_dB property in itself for the sound level.
- Usage:**
  - Plays different sounds based on the analog input value. Can be checked for the sound property SOUND\_dB by sensors to detect sound occurring.
- Direct Control:**
  - N/A
- Local Control:**
  - Connect to an analog slot to read and play sounds based on the value read.
- Remote Control:**
  - N/A
- Data Specifications:**
  - Input Slot: A0
- Example:**
  - Connect Potentiometer to Speaker using an MCU board. Turn the knob to play three different sounds.

### Specificații:

Play sounds.

#### Features:

- Analog input from 0 to 255 converted to predefined sounds.
- Sets a SOUND\_dB property in itself for the sound level.

#### Usage:

- Plays different sounds based on the analog input value. Can be checked for the sound property SOUND\_dB by sensors to detect sound occurring.

#### Direct Control:

- N/A

#### Local Control:

- Connect to an analog slot to read and play sounds based on the value read.

#### Remote Control:

- N/A

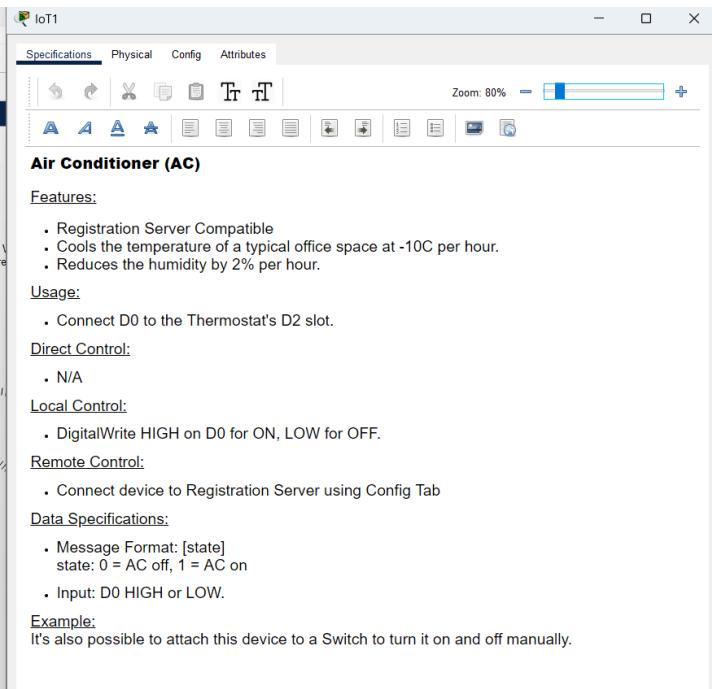
#### Data Specifications:

- Input Slot: A0

#### Example:

- Connect Potentiometer to Speaker using an MCU board. Turn the knob to play three different sounds.

## 2. Aer condiționat



The screenshot shows the IoT1 software interface with the 'Specifications' tab selected. On the left, there is a thumbnail image of an air conditioner unit labeled 'Air Conditioner IoT1'. The main panel displays the following information:

- Air Conditioner (AC)**
- Features:**
  - Registration Server Compatible
  - Cools the temperature of a typical office space at -10C per hour.
  - Reduces the humidity by 2% per hour.
- Usage:**
  - Connect D0 to the Thermostat's D2 slot.
- Direct Control:**
  - N/A
- Local Control:**
  - DigitalWrite HIGH on D0 for ON, LOW for OFF.
- Remote Control:**
  - Connect device to Registration Server using Config Tab
- Data Specifications:**
  - Message Format: [state]  
state: 0 = AC off, 1 = AC on
  - Input: D0 HIGH or LOW.
- Example:**

It's also possible to attach this device to a Switch to turn it on and off manually.

### Specificații:

#### Features:

- Registration Server Compatible
- Cools the temperature of a typical office space at -10C per hour.
- Reduces the humidity by 2% per hour.

#### Usage:

- Connect D0 to the Thermostat's D2 slot.

#### Direct Control:

N/A

#### Local Control:

- DigitalWrite HIGH on D0 for ON, LOW for OFF.

#### Remote Control:

- Connect device to Registration Server using Config Tab.

#### Data Specifications:

- Message Format: [state]  
state: 0 = AC off, 1 = AC on
- Input: D0 HIGH or LOW.

#### Example:

- It's also possible to attach this device to a Switch to turn it on and off manually.

### 3. Alarmă



**IoT2**

Specifications Attributes

Zoom: 100%

**Alarm**

Features:

- Default LOW.
- Alarm is triggered with a value of HIGH, and its light turns red.

Usage:

- N/A

Direct Control:

- N/A

Local Control:

- Connect device to MCU or SBC. Use the "digitalWrite" API per Data Specifications.

Remote Control:

- N/A

Data Specifications:  
Message Format: [state]  
state: HIGH or 1 = on, LOW or 0 = off

Example:  
Work with a trip sensor and an MCU to set off Siren.

#### Specificații:

##### Features:

- Default LOW.
- Alarm is triggered with a value of HIGH, and its light turns red.

##### Usage:

- N/A

##### Direct Control:

- N/A

##### Local Control:

- Connect device to MCU or SBC. Use the "digitalWrite" API per Data Specifications.

##### Remote Control:

- N/A

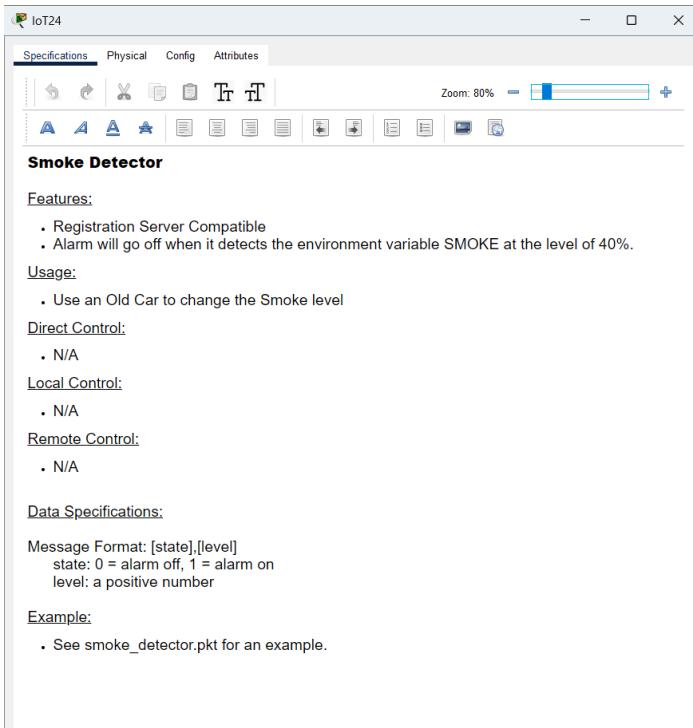
##### Data Specifications:

- Message Format: [state]  
state: HIGH or 1 = on, LOW or 0 = off

##### Example:

- Work with a trip sensor and an MCU to set off Siren.

## 4. Detector de fum



The screenshot shows the IoT24 software interface with the following details:

- Specifications Tab:** The active tab, showing the Smoke Detector's features, usage, direct control, local control, remote control, and data specifications.
- Features:**
  - Registration Server Compatible
  - Alarm will go off when it detects the environment variable SMOKE at the level of 40%.
- Usage:**
  - Use an Old Car to change the Smoke level
- Direct Control:**
  - N/A
- Local Control:**
  - N/A
- Remote Control:**
  - N/A
- Data Specifications:**

Message Format: [state],[level]  
state: 0 = alarm off, 1 = alarm on  
level: a positive number
- Example:**
  - See smoke\_detector.pkt for an example.

**Smoke Detector IoT24**

### Specificații:

#### Features:

- Registration Server Compatible
- Alarm will go off when it detects the environment variable SMOKE at the level of 40%.

#### Usage:

- Use an Old Car to change the Smoke level

#### Direct Control:

- N/A

#### Local Control:

- N/A

#### Remote Control:

- N/A

#### Data Specifications:

- Message Format: [state], [level]  
state: 0 = alarm off, 1 = alarm on  
level: a positive number

#### Example:

- See smoke\_detector.pkt for an example.

## 5. WebCam



IoT2

Specifications I/O Config Physical Config Thing Editor Programming Attributes

Edit Format Templates

Zoom: 70%

**Webcam**  
A camera device that records and sends data

Features:

- Registration Server Compatible
- Off
- On
- Video recording

Usage:

- N/A

Direct Control:

- ALT-click to interact

Local Control:

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

Remote Control:

- Connect device to Registration Server using Config Tab

Data Specifications:  
Message Format: [state]  
state: 0 = off, 1 = on

Example:

- Use motion\_detector.pkt sample file.

### Specificații:

A camera device that records and sends data

### Features:

- Registration Server Compatible
- Off
- On
- Video recording

### Usage:

- N/A

### Direct Control:

- ALT-click to interact

### Local Control:

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

### Remote Control:

- Connect device to Registration Server using Config Tab

### Data Specifications:

- Message Format: [state]  
state: 0 = off, 1 = on

### Example:

- Use motion\_detector(pkt sample file.

## 6. Laptop



Laptop0

Physical Config Desktop Programming Attributes

**Attributes:**

Name	Attribute
1 MTBF	26280
2 cost	1000
3 power source	0
4 rack units	2
5 wattage	60

**Properties:**

Property	Value
1 PROGRAMMING_EDITING_DIR	

### Specificații:

#### Name | Attribute

1. MTBF | 26280
2. Cost | 1000
3. Power Source | 0
4. Rack Units | 2
5. Wattage | 60

#### Properties: Property | Value

1. PROGRAMMING\_EDITING\_DIR | (No value shown)

## 7. Umidificator



Humidifier  
IoT17

The screenshot shows the IoT17 software interface with the 'Specifications' tab selected. The device is identified as a 'Humidifier'. Key sections include:

- Features:**
  - Registration Server Compatible
  - Off
  - On
  - Increases Humidity
- Usage:**
  - N/A
- Direct Control:**
  - ALT-click to interact
- Local Control:**
  - Connect device to MCU/SBC/Thing. Use the "customWrite" or digitalWrite API per Data Specifications
- Remote Control:**
  - Connect device to Registration Server using Config Tab
- Data Specifications:**
  - Input Slot: D0
  - Message Format: [state]  
state: 0 = off, 1 = on
- Example:**
  - N/A

### Specificații:

A Humidifier.

#### Features:

- Registration Server Compatible
- Off
- On
- Increases Humidity

#### Usage:

- N/A

#### Direct Control:

- ALT-click to interact

#### Local Control:

- Connect device to MCU/SBC/Thing. Use the "customWrite" or digitalWrite API per Data Specifications

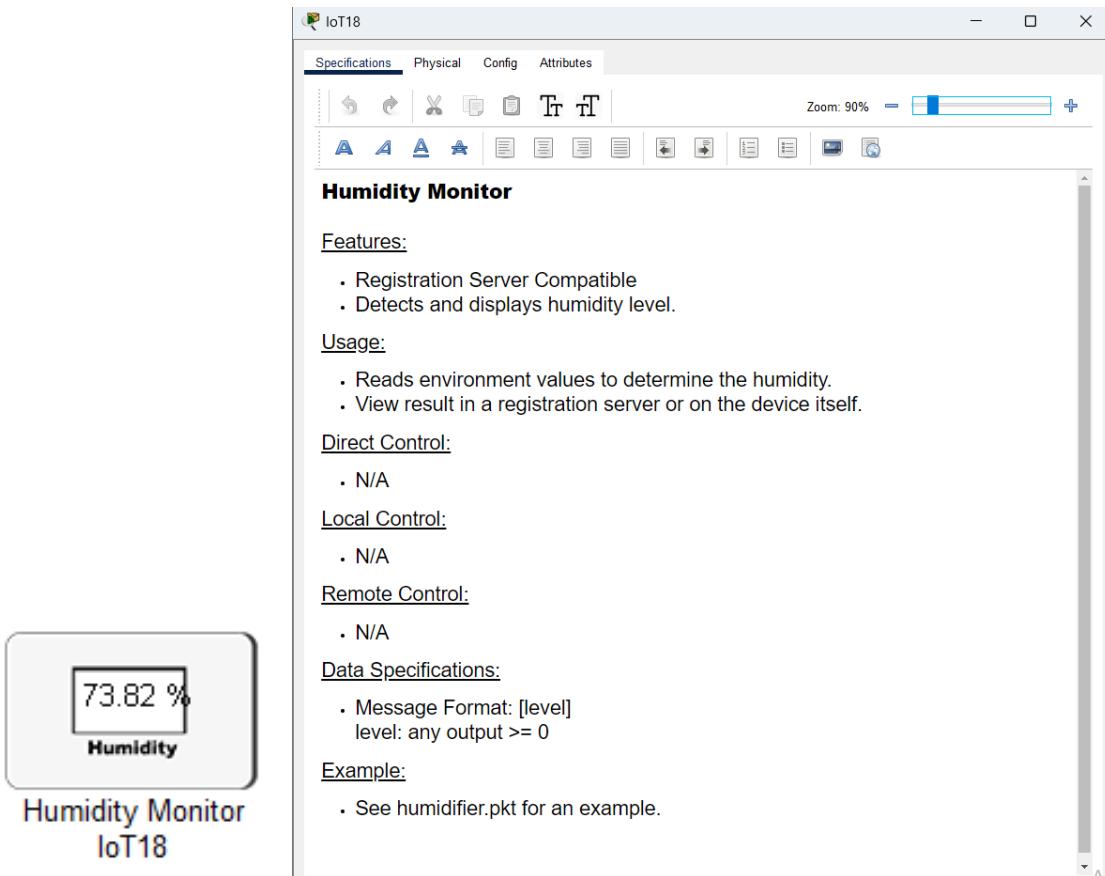
#### Remote Control:

- Connect device to Registration Server using Config Tab

#### Data Specifications:

- Input Slot: D0
- Message Format: [state]  
state: 0 = off, 1 = on

## 8. Dispozitiv umiditate



### Specificații:

#### Features:

- Registration Server Compatible
- Detects and displays humidity level.

#### Usage:

- Reads environment values to determine the humidity.
- View result in a registration server or on the device itself.

#### Direct Control:

- N/A

#### Local Control:

- N/A

#### Remote Control:

- N/A

#### Data Specifications:

- Message Format: [level]  
level: any output  $\geq 0$

## 9. Printer



Printer-PT  
Printer0

The screenshot shows a software interface for managing a printer. The title bar says "Printer0". Below it, there are three tabs: "Physical", "Config", and "Attributes", with "Attributes" being the active tab. Under "Attributes", there is a table with columns "Name" and "Attribute". The table contains five rows of data:

Name	Attribute
1 MTBF	17520
2 cost	1500
3 power source	0
4 rack units	3
5 wattage	30

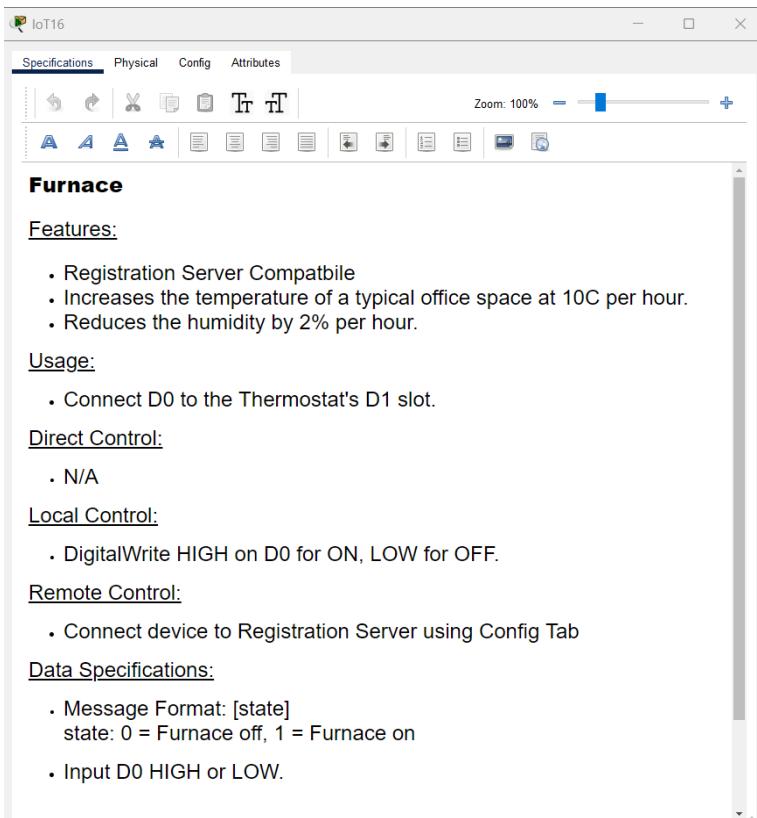
Below the table, under "Properties", there is a table with columns "Property" and "Value". This table is currently empty. At the bottom right of the window is a "Refresh" button.

### Specificații:

#### Attributes:

1. **MTBF:** 17520
2. **Cost:** 1500
3. **Power Source:** 0
4. **Rack Units:** 3
5. **Wattage:** 30

## 10. Calorifer



The screenshot shows the IoT16 software interface with the 'Furnace' device selected. The top menu bar includes 'Specifications', 'Physical', 'Config', and 'Attributes'. Below the menu is a toolbar with various icons. The main area displays the 'Furnace' configuration with sections for 'Features', 'Usage', 'Direct Control', 'Local Control', 'Remote Control', and 'Data Specifications'. A small image of a white vertical radiator is shown on the left.

**Furnace**

Features:

- Registration Server Compatible
- Increases the temperature of a typical office space at 10C per hour.
- Reduces the humidity by 2% per hour.

Usage:

- Connect D0 to the Thermostat's D1 slot.

Direct Control:

- N/A

Local Control:

- DigitalWrite HIGH on D0 for ON, LOW for OFF.

Remote Control:

- Connect device to Registration Server using Config Tab

Data Specifications:

- Message Format: [state]  
state: 0 = Furnace off, 1 = Furnace on
- Input D0 HIGH or LOW.

### Specificații:

#### Features:

- Registration Server Compatible
- Increases the temperature of a typical office space at 10C per hour.
- Reduces the humidity by 2% per hour.

#### Usage:

- Connect D0 to the Thermostat's D1 slot.

#### Direct Control:

- N/A

#### Local Control:

- DigitalWrite HIGH on D0 for ON, LOW for OFF.

#### Remote Control:

- Connect device to Registration Server using Config Tab

#### Data Specifications:

- Message Format: [state]  
state: 0 = Furnace off, 1 = Furnace on  
Input D0 HIGH or LOW.

## 11. Detector mișcare



Motion Detector  
IoT21

### Specificații:

#### Features:

- Registration Server Compatible
- Detects motion from mouse movement.
- Automatically deactivates after 5 seconds without any mouse movement.

#### Usage:

- N/A

#### Direct Control:

- Alt-mouse move to interact.

#### Local Control:

- N/A

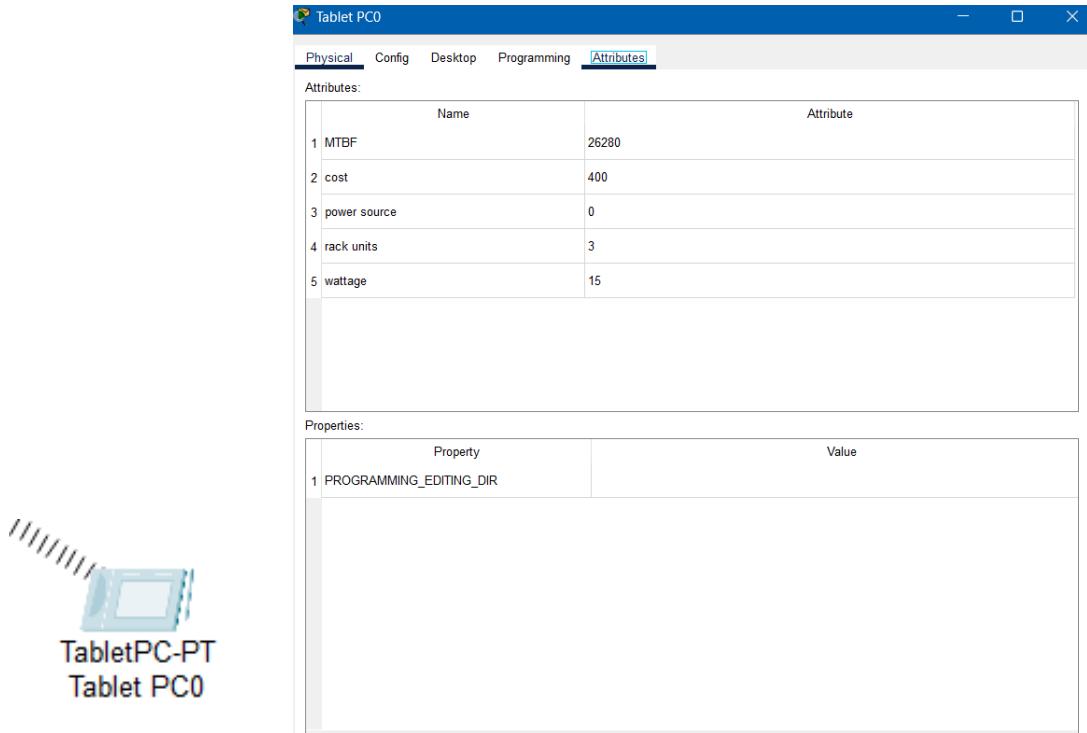
#### Remote Control:

- N/A

#### Data Specifications:

- Message Format: [state]  
state: HIGH = activated, LOW = inactive

## 12. Tableta

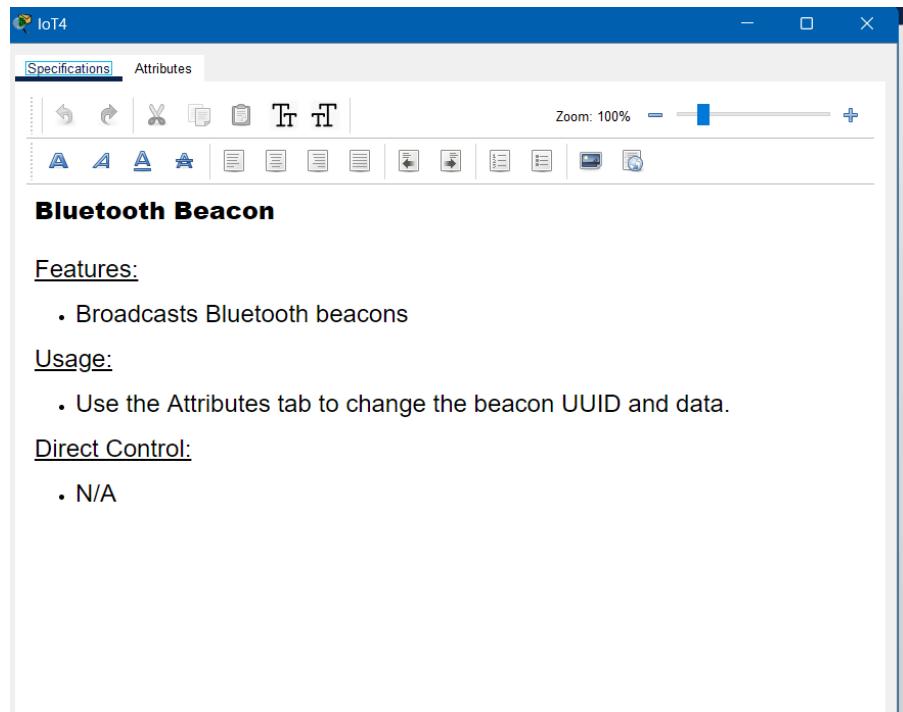


### Specificații:

#### Attributes:

1. **MTBF:** 26280
2. **Cost:** 400
3. **Power Source:** 0
4. **Rack Units:** 3
5. **Wattage:** 6

## 13. Bluetooth Beacon



### Specificații:

#### Features:

- Broadcasts Bluetooth beacons

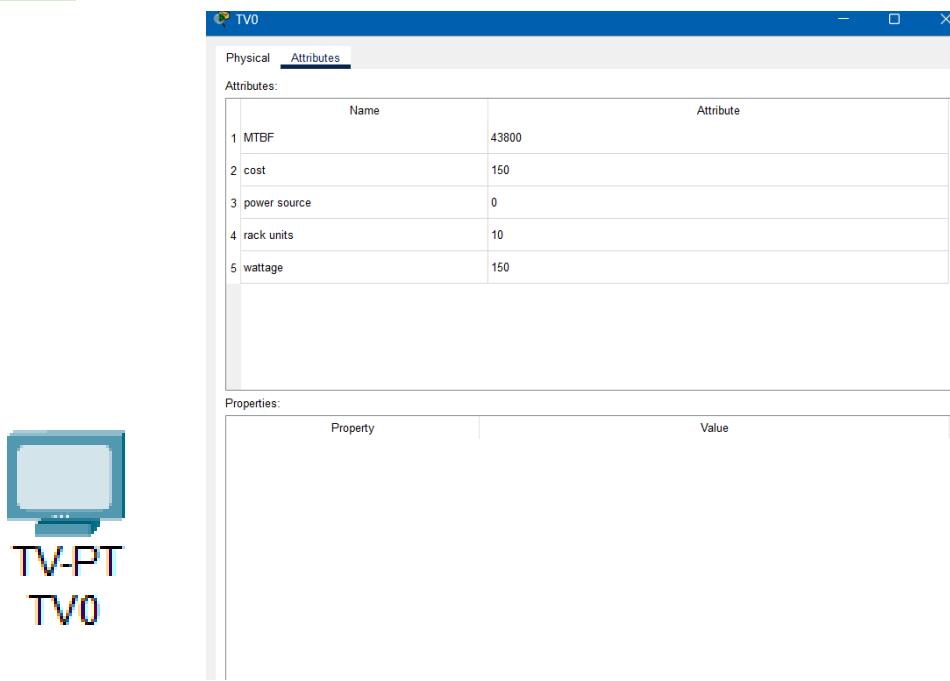
#### Usage:

- Use the Attributes tab to change the beacon UUID and data.

#### Direct Control:

- N/A

## 14. TV

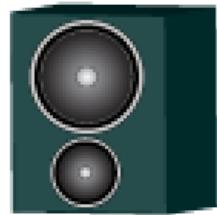


### Specificații:

#### Attributes:

1. **MTBF:** 43800
2. **Cost:** 150
3. **Power Source:** 0
4. **Rack Units:** 10
5. **Wattage:** 150

## 16. Bluetooth Speaker



Bluetooth Speaker  
IoT4

The screenshot shows the IoT4 software interface with the 'Specifications' tab selected. The main content area displays the following information:

- Bluetooth Speaker**: Plays sound through Bluetooth from a Portable Music Player.
- Features:**
  - Bluetooth compatible
  - Registration Server Compatible
- Usage:**
  - Pair a Bluetooth Speaker with the Portable Music Player through Bluetooth using the Config tab. Enable the Portable Music Player by pressing ALT-click. The Portable Music Player will then send data to the Bluetooth Speaker through Bluetooth to play white noise music.
- Direct Control:**
  - N/A
- Local Control:**
  - N/A
- Remote Control:**
  - N/A
- Data Specifications:**
  - N/A
- Example:**
  - See Usage section.

### Specificații:

Plays sound through Bluetooth from a Portable Music Player.

#### Features:

- Bluetooth compatible
- Registration Server Compatible

#### Usage:

- Pair a Bluetooth Speaker with the Portable Music Player through Bluetooth using the Config tab.
- Enable the Portable Music Player by pressing ALT-click.
- The Portable Music Player will then send data to the Bluetooth Speaker through Bluetooth to play white noise music.

#### Direct Control:

- N/A

#### Local Control:

- N/A

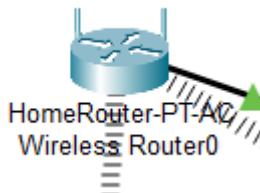
#### Remote Control:

- N/A

#### Data Specifications:

- N/A

## 16. Router



Wireless Router0

Physical Config GUI Attributes

Attributes:

Name	Attribute
1 MTBF	26000
2 cost	100
3 power source	0
4 rack units	2
5 wattage	50

Properties:

Property	Value

### Specificații:

#### Attributes:

1. **MTBF:** 26000
2. **Cost:** 100
3. **Power Source:** 0
4. **Rack Units:** 2
5. **Wattage:** 50

## 17. Led



IoT4

Specifications Attributes

**RGB LED**  
GRB Common Cathode

Features:

- Color: different color based on RGB values
- High Reliability

Usage:

- Connect to MCU/SBC to Slot A0, A1, and A2
- Analog input of 1023 will be mapped to 255 and 0 will be mapped to 0

Direct Control:

- N/A

Local Control:

- Connect device to an MCU. From MCU, use "analogWrite" to adjust the LED color.

Remote Control:

- N/A

Data Specifications:

- Input Slot:
  - A0 [value range 0-1023] - Red
  - A1 [value range 0-1023] - Green
  - A2 [value range 0-1023] - Blue

Example:

- N/A

## Specificații:

### GRB Common Cathode

#### Features:

- Color: different color based on RGB values
- High Reliability

#### Usage:

- Connect to MCU/SBC to Slot A0, A1, and A2
- Analog input of 1023 will be mapped to 255 and 0 will be mapped to 0

#### Direct Control:

- N/A

#### Local Control:

- Connect device to an MCU. From MCU, use "analogWrite" to adjust the LED color.

#### Remote Control:

- N/A

#### Data Specifications:

##### • Input Slot:

- A0 [value range 0-1023] - Red
- A1 [value range 0-1023] - Green
- A2 [value range 0-1023] - Blue

## 18. Lampă



The screenshot shows the IoT20 software interface with the 'Light' component selected. The top navigation bar includes tabs for 'Specifications', 'Physical', 'Config', and 'Attributes'. The 'Specifications' tab is active, displaying the following details:

**Light**  
A lamp that can be turned on or off.

**Features:**

- Registration Server Compatible
- Off
- Dim
- On
- Emits light to the environment

**Usage:**

- N/A

**Direct Control:**

- ALT-click to interact

**Local Control:**

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

**Remote Control:**

- Connect device to Registration Server using Config Tab

**Data Specifications:**  
Message Format: [state]  
state: 0 = off, 1 = dim, 2 = on

**Example:**

- Connect an MCU to Light. Cycle through 0, 1, and 2 to change Light state.

## Specificații:

A lamp that can be turned on or off.

### Features:

- Registration Server Compatible
- Off
- Dim
- On
- Emits light to the environment

### Usage:

- N/A

### Direct Control:

- ALT-click to interact

### Local Control:

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

### Remote Control:

- Connect device to Registration Server using Config Tab

### Data Specifications:

- Message Format: [state]  
state: 0 = off, 1 = dim, 2 = on

### Example:

- Connect an MCU to Light. Cycle through 0, 1, and 2 to change Light state.

## 19. Ușă

The screenshot shows the IoT15 software interface with the 'Door' component selected. The top navigation bar includes tabs for 'Specifications', 'Physical', 'Config', and 'Attributes'. The 'Specifications' tab is active, displaying the following details:

**Door**  
Open / Close / Unlock / Lock

**Features:**

- Registration Server Compatible
- Ability to vent Carbon Dioxide and Carbon Monoxide

**Usage:**

- Connect to the Door from SBC/MCU/Thing with IoT Custom Cable
- Use customWrite function to control the door and lock

**Direct Control:**

- ALT-click on keyhole to lock/unlock
- ALT-click on door to open/close

**Local Control:**

- Connect device to SBC/MCU/Thing. Use the "customWrite" API per Data Specifications

**Remote Control:**

- Connect device to Registration Server using Config Tab

NOTE: opening and closing the door is not remote controllable

**Data Specifications:**  
Message Format: [door].[lock]  
door: 0 = closed, 1 = open, -1 = don't care  
lock: 0 = unlock, 1 = lock, -1 = don't care

**Example:**  
Connect a SBC to the door, send a customWrite and open the door to vent the Carbon Dioxide and Carbon Monoxide level

On the left side of the interface, there is a small icon of a door with a keyhole, labeled "Door IoT15".

## **Specificații:**

Open / Close / Unlock / Lock

## **Features:**

- Registration Server Compatible
- Ability to vent Carbon Dioxide and Carbon Monoxide

## **Usage:**

- Connect to the Door from SBC/MCU/Thing with IoT Custom Cable
- Use customWrite function to control the door and lock

## **Direct Control:**

- ALT-click on keyhole to lock/unlock
- ALT-click on door to open/close

## **Local Control:**

- Connect device to SBC/MCU/Thing. Use the "customWrite" API per Data Specifications

## **Remote Control:**

- Connect device to Registration Server using Config Tab
- **NOTE:** Opening and closing the door is not remote controllable

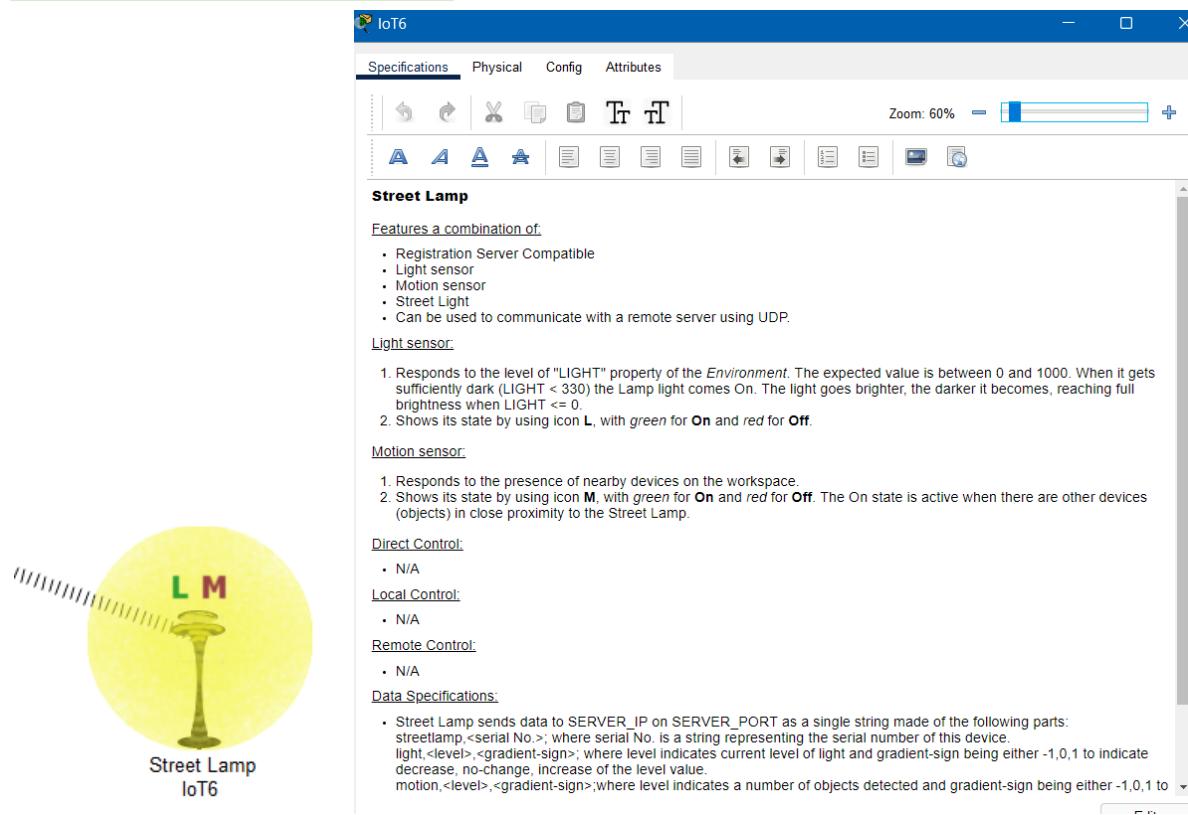
## **Data Specifications:**

- **Message Format:** [door], [lock]  
door: 0 = closed, 1 = open, -1 = don't care  
lock: 0 = unlock, 1 = lock, -1 = don't care

## **Example:**

- Connect an SBC to the door, send a customWrite and open the door to vent the Carbon Dioxide and Carbon Monoxide level.

## 20. Lampa Pentru Intrare



The screenshot shows the IoT6 software interface with the 'Specifications' tab selected. The main content area displays the 'Street Lamp' device. It includes a description of the device's features, a diagram, and various control and configuration options.

**Features a combination of:**

- Registration Server Compatible
- Light sensor
- Motion sensor
- Street Light
- Can be used to communicate with a remote server using UDP.

**Light sensor:**

1. Responds to the level of "LIGHT" property of the *Environment*. The expected value is between 0 and 1000. When it gets sufficiently dark (LIGHT < 330) the Lamp light comes On. The light goes brighter, the darker it becomes, reaching full brightness when LIGHT <= 0.
2. Shows its state by using icon **L**, with **green** for **On** and **red** for **Off**.

**Motion sensor:**

1. Responds to the presence of nearby devices on the workspace.
2. Shows its state by using icon **M**, with **green** for **On** and **red** for **Off**. The **On** state is active when there are other devices (objects) in close proximity to the Street Lamp.

**Direct Control:**

- N/A

**Local Control:**

- N/A

**Remote Control:**

- N/A

**Data Specifications:**

- Street Lamp sends data to SERVER\_IP on SERVER\_PORT as a single string made of the following parts:  
streetlamp,<serial No.>; where serial No. is a string representing the serial number of this device.  
light,<level>,<gradient-sign>; where level indicates current level of light and gradient-sign being either -1,0,1 to indicate decrease, no-change, increase of the level value.  
motion,<level>,<gradient-sign>; where level indicates a number of objects detected and gradient-sign being either -1,0,1 to

### Specificații:

#### Features a combination of:

- Registration Server Compatible
- Light sensor
- Motion sensor
- Street Light
- Can be used to communicate with a remote server using UDP

#### Light sensor:

1. Responds to the level of "LIGHT" property of the *Environment*. The expected value is between 0 and 1000. When it gets sufficiently dark (LIGHT < 330), the Lamp light turns **On**. The light gets brighter as it gets darker, reaching full brightness when LIGHT <= 0.
2. Shows its state using icon **L**, with **green** for **On** and **red** for **Off**.

#### Motion sensor:

1. Responds to the presence of nearby devices on the workspace.
2. Shows its state using icon **M**, with **green** for **On** and **red** for **Off**. The **On** state is active when there are other devices (objects) in close proximity to the Street Lamp.

#### Direct Control:

- N/A

#### Local Control:

- N/A

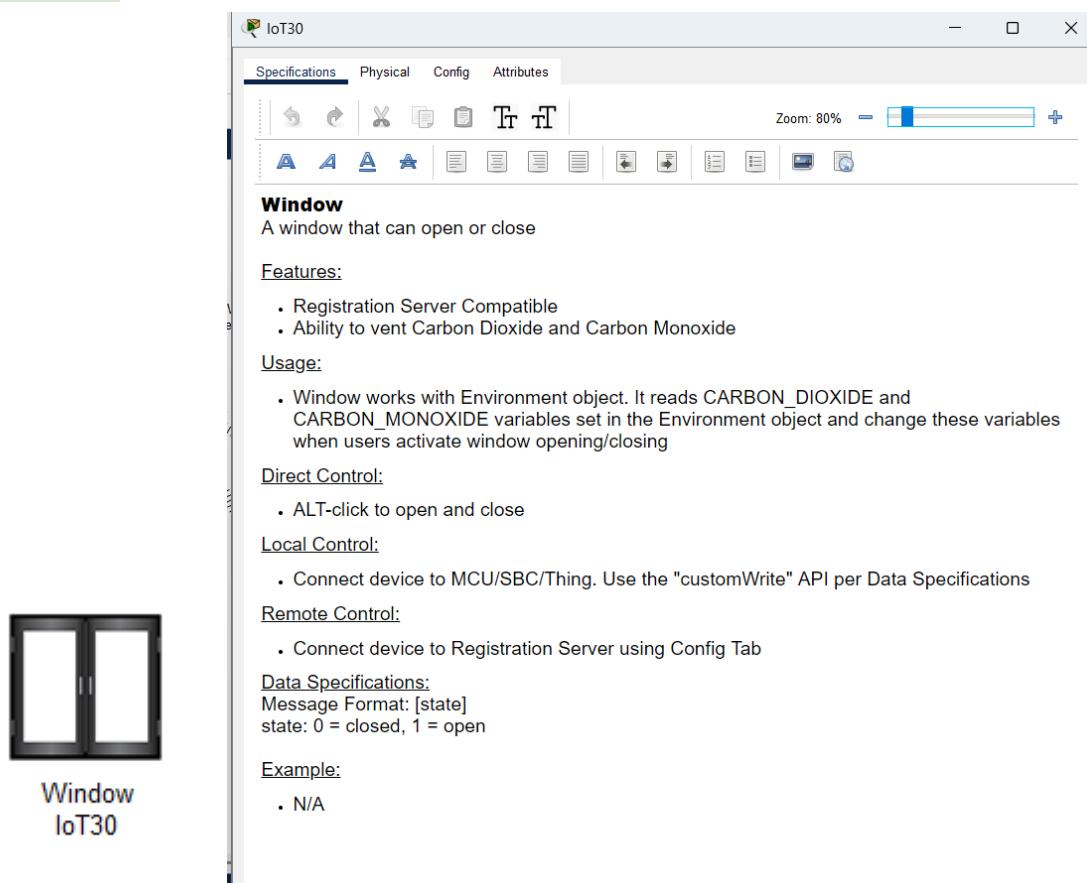
## Remote Control:

- N/A

## Data Specifications:

- Street Lamp sends data to **SERVER\_IP** on **SERVER\_PORT** as a single string made of the following parts:
  - **streetlamp.<serial No.>**; where *serial No.* is a string representing the serial number of this device.
  - **light,<level>,<gradient-sign>**; where *level* indicates current light level and *gradient-sign* can be **-1, 0, or 1** to indicate **decrease, no change, or increase** of the level.
  - **motion,<level>,<gradient-sign>**; where *level* indicates the number of objects detected and *gradient-sign* is either **-1, 0, or 1**.

## 21. Geam



## Specificații:

A window that can open or close

## Features:

- Registration Server Compatible
- Ability to vent Carbon Dioxide and Carbon Monoxide

### **Usage:**

- Window works with *Environment* object. It reads **CARBON\_DIOXIDE** and **CARBON\_MONOXIDE** variables set in the *Environment* object and changes these variables when users activate window opening/closing.

### **Direct Control:**

- ALT-click to open and close

### **Local Control:**

- Connect device to MCU/SBC/Thing. Use the "**customWrite**" API per Data Specifications

### **Remote Control:**

- Connect device to Registration Server using Config Tab

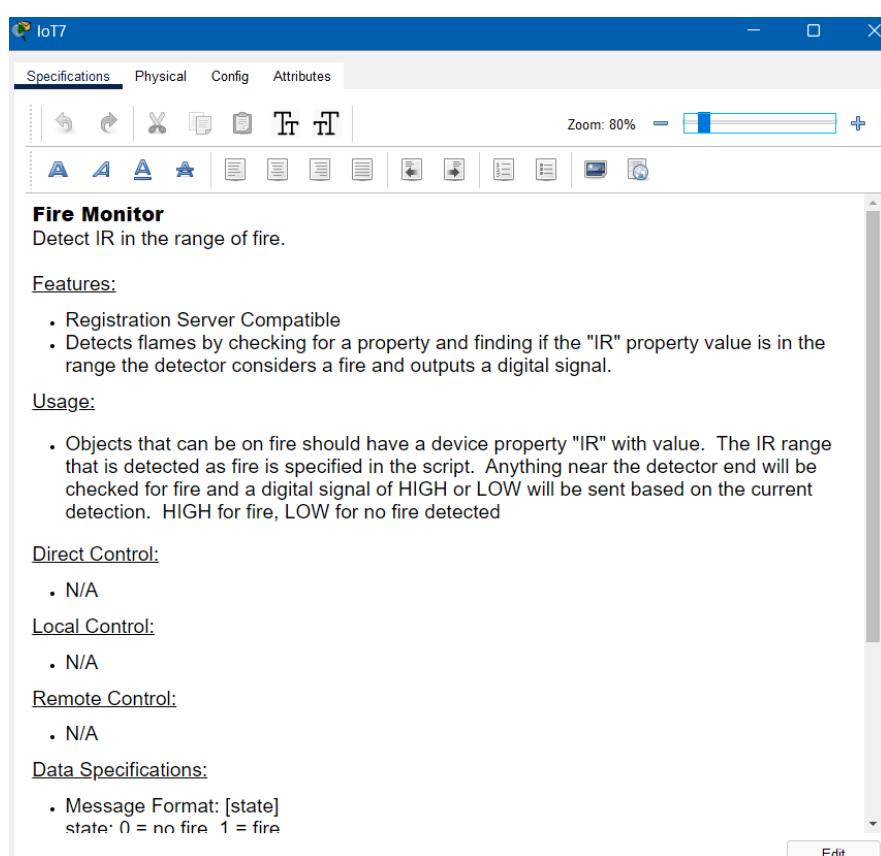
### **Data Specifications:**

- **Message Format:** [state]  
state: **0 = closed, 1 = open**

## **22. Fire Monitor**



Fire Monitor  
IoT7



The screenshot shows the IoT7 software interface with the "Specifications" tab selected. The main area displays the following information for the Fire Monitor:

- Fire Monitor**: Detect IR in the range of fire.
- Features:**
  - Registration Server Compatible
  - Detects flames by checking for a property and finding if the "IR" property value is in the range the detector considers a fire and outputs a digital signal.
- Usage:**
  - Objects that can be on fire should have a device property "IR" with value. The IR range that is detected as fire is specified in the script. Anything near the detector end will be checked for fire and a digital signal of HIGH or LOW will be sent based on the current detection. HIGH for fire, LOW for no fire detected
- Direct Control:**
  - N/A
- Local Control:**
  - N/A
- Remote Control:**
  - N/A
- Data Specifications:**
  - Message Format: [state]  
state: 0 = no fire 1 = fire

### **Specificații:**

Detect IR in the range of fire.

### **Features:**

- Registration Server Compatible

- Detects flames by checking for a property and finding if the "IR" property value is in the range the detector considers a fire and outputs a digital signal.

#### Usage:

- Objects that can be on fire should have a device property "IR" with a value. The IR range that is detected as fire is specified in the script.
- Anything near the detector end will be checked for fire, and a digital signal of **HIGH** or **LOW** will be sent based on the current detection.
  - **HIGH** for fire, **LOW** for no fire detected.

#### Direct Control:

- N/A

#### Local Control:

- N/A

#### Remote Control:

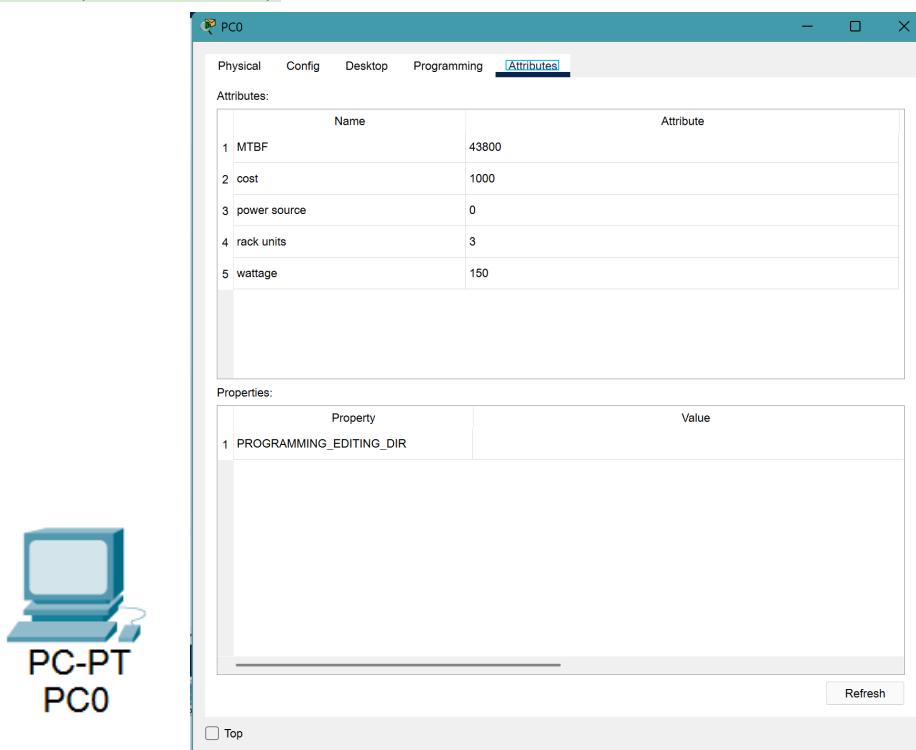
- N/A

#### Data Specifications:

- **Message Format:** [state]

state: **0 = no fire, 1 = fire**

## 23. PC (Calculator)

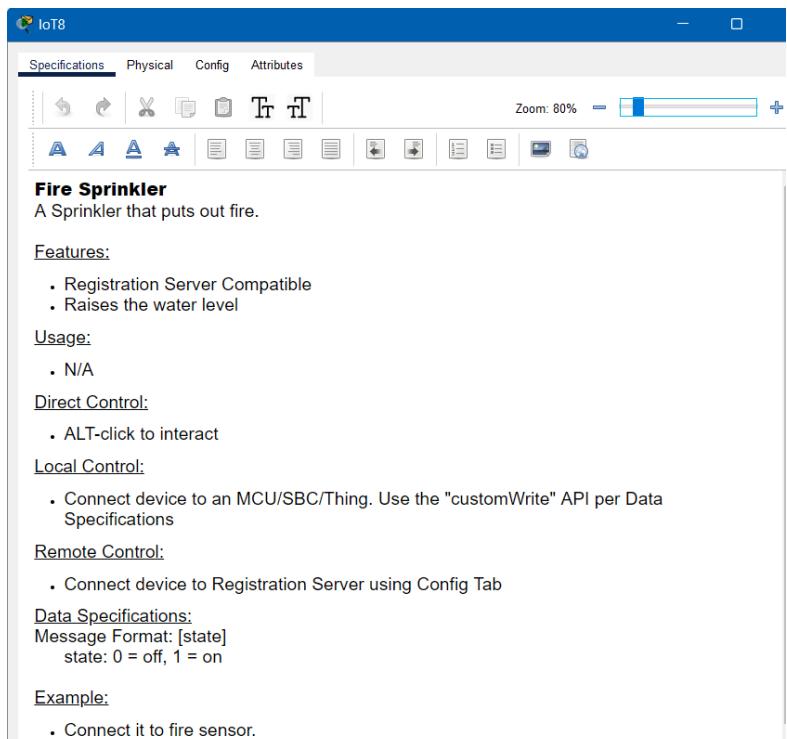


## Specificații:

### Attributes:

1. **MTBF:** 43800
2. **Cost:** 1000
3. **Power Source:** 0
4. **Rack Units:** 3
5. **Wattage:** 150

## 24. Fire Sprinkler



## Specificații:

A Sprinkler that puts out fire.

### Features:

- Registration Server Compatible
- Raises the water level

### Usage:

- N/A

### Direct Control:

- ALT-click to interact

### Local Control:

- Connect device to an MCU/SBC/Thing. Use the "**customWrite**" API per Data Specifications

## Remote Control:

- Connect device to Registration Server using Config Tab

## Data Specifications:

- **Message Format:** [state]  
state: **0 = off, 1 = on**

## Example:

- Connect it to a fire sensor.

## 25. Baterie

**Battery**

Features:

- Registration Server Compatible
- Shows the percentage of charge left in the battery

Usage:

- Send power to other devices

Direct Control:

- N/A

Local Control:

- N/A

Remote Control:

- N/A

Data Specifications:  
Inputs and outputs use logarithmic scale in order to accommodate power values up to 100 MWh (+/-20%).

**Input Slot: D0**  
Receives the power from a device such as the solar panel

**Output Slots: All other slots**  
Sends power to the receiving device

Example:

- Connect slot 0 to a device that supplies power such as the solar panel. Doing this allows the device supplying power to charge the battery.
- Connect slot 1 to a power consuming device. The battery automatically detects the appropriate amount of power to send to the device and sends only if there is enough energy stored in the battery.
- Edit the programming of the receiving device to require power from the battery. A simple statement such as:

```
if (analogRead(powerPin) > 0) {  
    powerOn = true;  
}  
else {  
    powerOn = false;  
}
```

## Specificații:

### Features:

- Registration Server Compatible
- Shows the percentage of charge left in the battery

### Usage:

- Send power to other devices

### Direct Control:

- N/A

### Local Control:

- N/A

### Remote Control:

- N/A

### Data Specifications:

- Inputs and outputs use a logarithmic scale to accommodate power values up to **100 MWh (+/-20%)**.
- **Input Slot: D0** – Receives power from a device such as a solar panel.
- **Output Slots: All other slots** – Sends power to the receiving device.

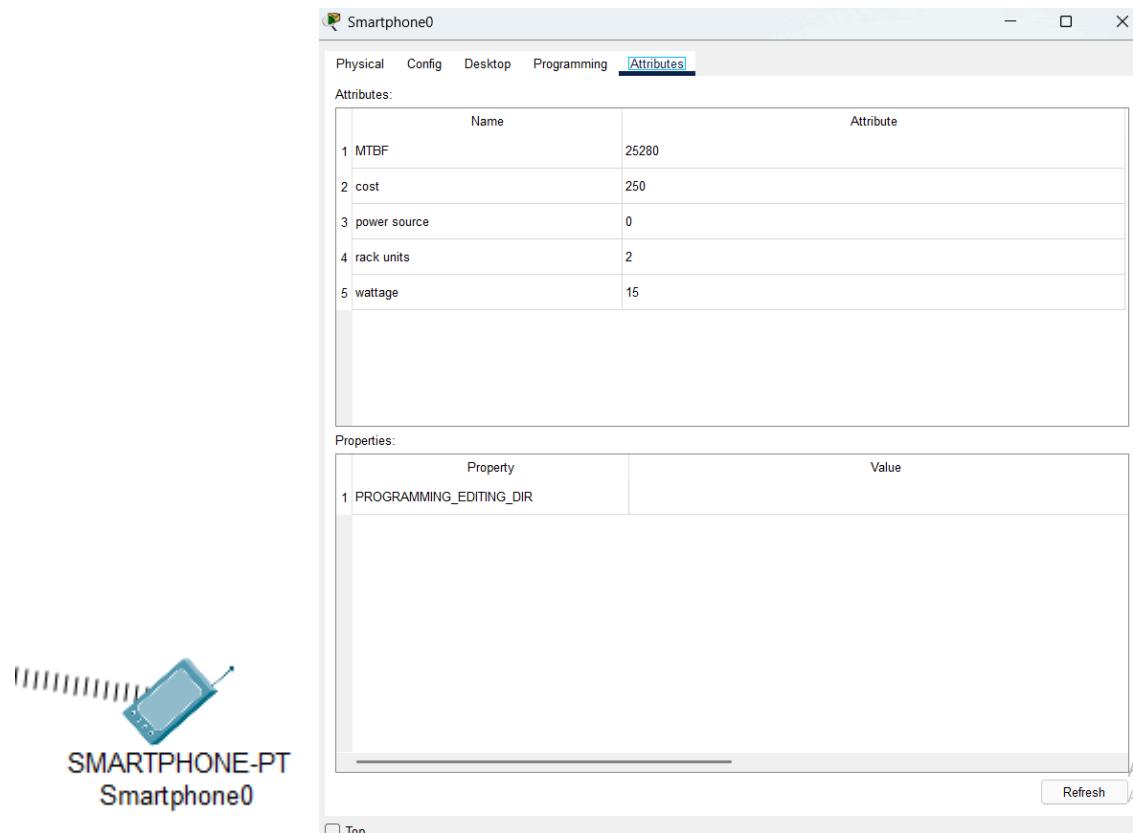
### Example:

- Connect **slot 0** to a device that supplies power, such as a solar panel, allowing it to charge the battery.
- Connect **slot 1** to a power-consuming device. The battery automatically detects the appropriate amount of power to send and does so only if there is enough energy stored.

Edit the programming of the receiving device to require power from the battery. A simple statement such as:

```
if (analogRead(powerPin) > 0) {  
    powerOn = true;  
} else {  
    powerOn = false;  
}
```

## 26. Smartphone

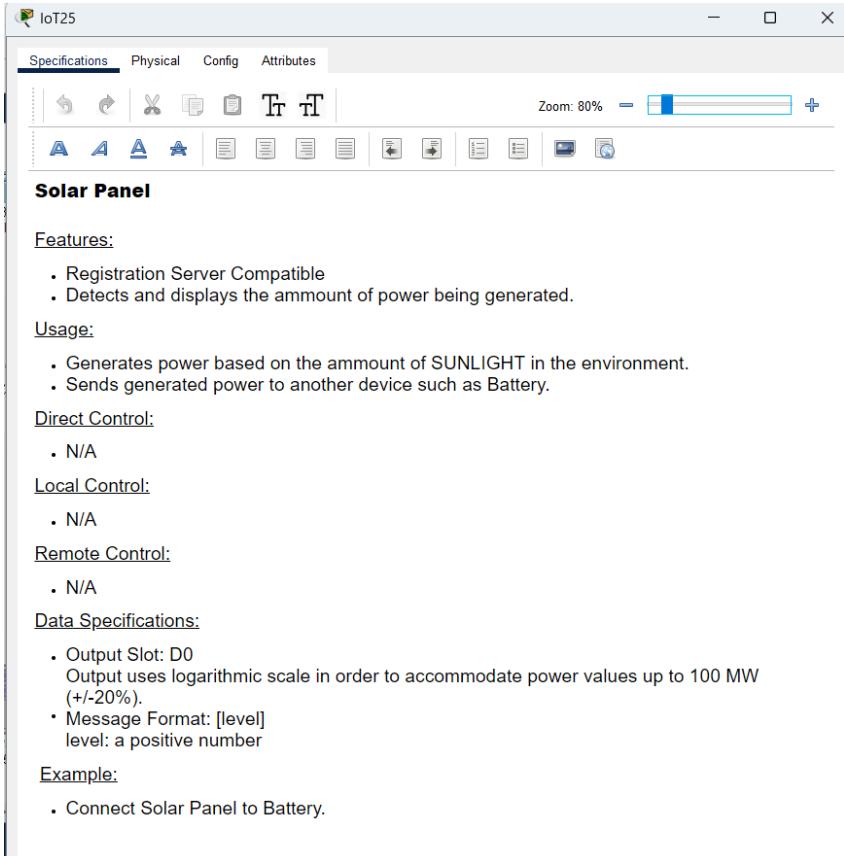


## Specificații:

### Attributes:

1. **MTBF:** 25280
2. **Cost:** 250
3. **Power Source:** 0
4. **Rack Units:** 2
5. **Wattage:** 15

## 27. Panou Solar



The screenshot shows the IoT25 software interface with the following details for the Solar Panel device:

- Features:**
  - Registration Server Compatible
  - Detects and displays the amount of power being generated.
- Usage:**
  - Generates power based on the amount of SUNLIGHT in the environment.
  - Sends generated power to another device such as Battery.
- Direct Control:**
  - N/A
- Local Control:**
  - N/A
- Remote Control:**
  - N/A
- Data Specifications:**
  - Output Slot: D0  
Output uses logarithmic scale in order to accommodate power values up to 100 MW (+/-20%).
  - Message Format: [level]  
level: a positive number
- Example:**
  - Connect Solar Panel to Battery.

**Solar Panel**  
**IoT25**

## Specificații:

### Features:

- Registration Server Compatible
- Detects and displays the amount of power being generated.

### Usage:

- Generates power based on the amount of **SUNLIGHT** in the environment.
- Sends generated power to another device such as a **Battery**.

### Direct Control:

- N/A

### **Local Control:**

- N/A

### **Remote Control:**

- N/A

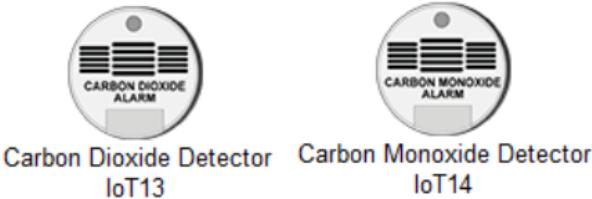
### **Data Specifications:**

- **Output Slot:** D0
- Output uses a logarithmic scale in order to accommodate power values up to **100 MW** (+/-20%).
- **Message Format:** [level]
  - level: a positive number

### **Example:**

- Connect Solar Panel to Battery.

## **28. Detector gaze**



**IoT14**

Specifications Physical Config Attributes

Carbon Monoxide Detector  
Detects the level of the carbon monoxide

**Features:**

- Registration Server Compatible
- Alarm will go off when it detects a Carbon Monoxide level of 20%

**Usage:**

- Use an Old Car to change the Carbon Monoxide level

**Direct Control:**

- N/A

**Local Control:**

- N/A

**Remote Control:**

- N/A

**Data Specifications:**

Message Format: [state],[level]  
state: 0 = alarm off, 1 = alarm on  
level: a positive number

**Example:**  
Start an Old Car to increase the Carbon Monoxide level, alarm will turn on when the level > 20%

### **Specificații:**

Detects the level of carbon monoxide.

### **Features:**

- Registration Server Compatible

- Alarm will go off when it detects a **Carbon Monoxide** level of **20%**

### Usage:

- Use an **Old Car** to change the Carbon Monoxide level

### Direct Control:

- N/A

### Local Control:

- N/A

### Remote Control:

- N/A

### Data Specifications:

- **Message Format:** [state],[level]
  - state: **0 = alarm off, 1 = alarm on**
  - level: **a positive number**

### Example:

- Start an **Old Car** to increase the **Carbon Monoxide** level; the alarm will turn on when the level > **20%**.

## 29. Thermostat

**Thermostat**

**Features:**

- Registration Server Compatible
- Off Mode
- Cooling Mode
- Heating Mode
- Auto Mode

**Usage:**

- Connect a Furnace to D1
- Connect an Air Conditioner (AC) to D2

**Direct Control:**

- ALT-click to interact

**Local Control:**

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

**Remote Control:**

- Connect the device to Registration Server using Config Tab

**Data Specifications:**

- Input Slot: D0  
Message Format: [state],[temperature],[autoCoolTemperature],[autoHeatTemperature]  
state:  
0 = off, 1 = cooling, 2 = heating, 3 = auto  
  
temperature:  
Current temperature in Celcius
- autoCoolTemperature:  
If in Auto mode and current temperature is >= autoCoolTemperature, it would start cooling
- autoHeatTemperature:  
If in Auto mode and current temperature is <= autoHeatTemperature, it would start heating
- Output Slot: D1  
Heating device such as Furnace
- Output Slot: D2  
Cooling device such as AC

**Example:**  
Work with a temperature sensor to show changes ranging from heating to cooling.

**Thermostat**  
IoT7

## **Specificații:**

### **Features:**

- Registration Server Compatible
- Off Mode
- Cooling Mode
- Heating Mode
- Auto Mode

### **Usage:**

- Connect a Furnace to D1
- Connect an Air Conditioner (AC) to D2

### **Direct Control:**

- ALT-click to interact

### **Local Control:**

- Connect device to MCU/SBC/Thing. Use the "customWrite" API per Data Specifications

### **Remote Control:**

- Connect the device to Registration Server using Config Tab

### **Data Specifications:**

- Input Slot: D0

Message Format: [state],[temperature],[autoCoolTemperature],[autoHeatTemperature]  
state:

0 = off, 1 = cooling, 2 = heating, 3 = auto

temperature:

Current temperature in Celcius

autoCoolTemperature:

If in Auto mode and current temperature is  $\geq$  autoCoolTemperature, it would start cooling

autoHeatTemperature:

If in Auto mode and current temperature is  $\leq$  autoHeatTemperature, it would start heating

- Output Slot: D1  
Heating device such as Furnace
- Output Slot: D2  
Cooling device such as AC

### **Example:**

- Work with a temperature sensor to show changes ranging from heating to cooling.

## 30. Trip Sensor



**Trip Sensor  
IoT10**

The screenshot shows the IoT10 software interface for the Trip Sensor. The top menu bar includes 'Specifications', 'Physical', 'Config', and 'Attributes'. Below the menu is a toolbar with various icons. The main content area is titled 'Trip Sensor' and contains the following information:

- Features:**
  - Registration Server Compatible
  - Activates when laser is tripped
- Usage:**
  - N/A
- Direct Control:**
  - ALT-mouse move to interact
- Local Control:**
  - N/A
- Remote Control:**
  - N/A
- Data Specifications:**

Message Format: [state]  
state: 0 = reset (LOW), 1 = tripped (HIGH)
- Example:**

Work with a siren to show tripping activating an alarm.

### Specificații:

Trip sensor is a type of motion detector that can detect movement across a laser beam.

#### Features:

- Registration Server Compatible
- Activates when laser is tripped

#### Usage:

- N/A

#### Direct Control:

- ALT-mouse move to interact

#### Local Control:

- N/A

#### Remote Control:

- N/A

#### Data Specifications:

- **Message Format:** [state]  
state: **0 = reset (LOW), 1 = tripped (HIGH)**

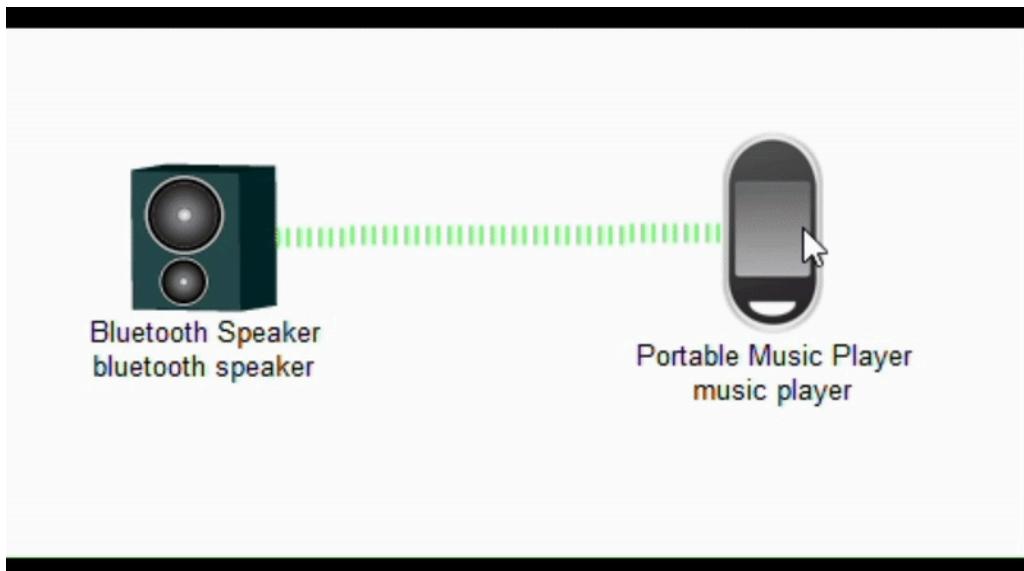
#### Example:

- Work with a siren to show tripping, activating an alarm

## — CONECTAREA OBIECTELOR —

Listening to Bluetooth Music<sup>1</sup>: **Bluetooth Speaker & Portable Music Player**

**Utilitate:** pentru ora de muzică, pentru competențe limbi străine audio



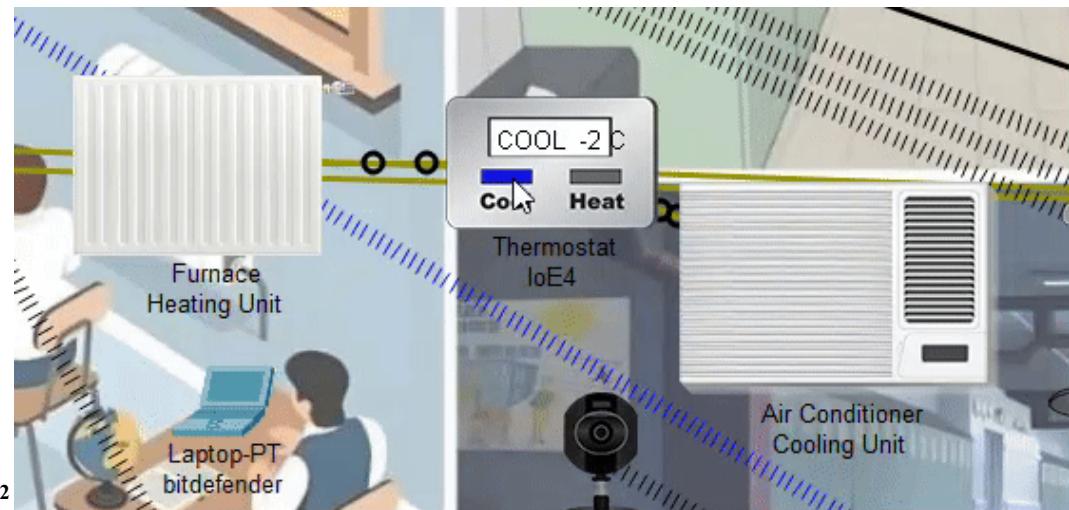
Pași:

- click **Bluetooth Speaker** (End Devices & Home)
- Config → Wireless0 → debifăm Port Status → Bluetooth → bifăm Port Status
- click **Portable Music Player** (End Devices & Home)
- Config → Wireless0 → debifăm Port Status → Bluetooth → bifăm Port Status → Discover (o să ne apară celălalt obiect, îl selectăm și apăsăm Pair → se conectează)
- apăsăm pe Portable Music Player folosind ALT-click

---

<sup>1</sup> <https://www.youtube.com/watch?v=ucs0DWLSdjo>

## Căldură & Aer condiționat: Furnace & Thermostat & Air Conditioner



**Utilitate:** Furnace și Air Conditioner sunt interdependente prin intermediul Termostatului: acesta „ascultă” temperatura dorită și le activează/dezactivează în funcție de nevoi

### Pasi:

- obiecte din (End Devices & Home)
- am utilizat pentru conectare cablul **Custom Cable**  , astfel Thermostat (D1) → Furnace (D0) 🔥 (activează încălzirea), Thermostat (D2) → Air Conditioner (D0) ❄️ (activează răcirea)



- la secțiunea **Programming**:
  - Thermostatul citește temperatura ambientală  
*temperature = Environment.get("Ambient Temperature");*
  - Thermostatul decide ce trebuie să facă  
*Dacă temperatura ≥ 20°C → Pornește AC (Cooling)  
Dacă temperatura ≤ 19°C → Pornește Furnace (Heating)  
Dacă este în modul „Auto”, comută între Cooling și Heating automat*
  - Dacă apăsăm click pe Thermostat, schimbă manual modul între:  
*OFF → COOL → HEAT → AUTO*

Cod principal pentru schimbarea temperaturii

Auto Mode:

```
if (temperature >= autoCoolTemp) {
```

---

<sup>2</sup> <https://www.youtube.com/watch?v=-nD7jKrsu0o&t=167s>

```

digitalWrite(COOLER_PIN, HIGH); // Activează AC (Cooling)
} else if (temperature <= autoHeatTemp) {
    digitalWrite(HEATER_PIN, HIGH); // Activează Furnace (Heating)
}

```

Schimbarea manuală a modului prin click:

```

function mouseEvent (pressed, x, y, firstPress) {
    if (firstPress) {
        state++;
        if (state >= 4) {
            state = 0;
        }
        setState(state);
    }
}

```

Dispozitive sunt conectate la „**Home Gateway**” și pot fi controlate prin **TabletPC-PT**<sup>3</sup>

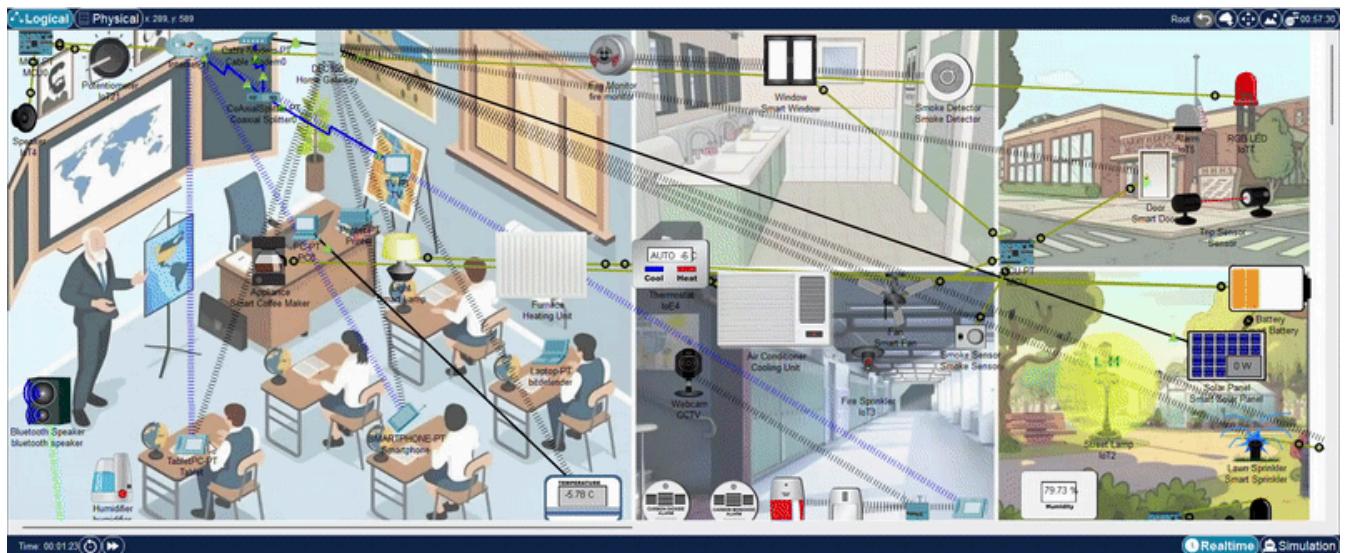



---

<sup>3</sup> <https://www.youtube.com/watch?v=9drBPsHAAww>

De asemenea dispozitivele pot fi controlate și prin **Web Browser** folosind **tableta**

<sup>4</sup>



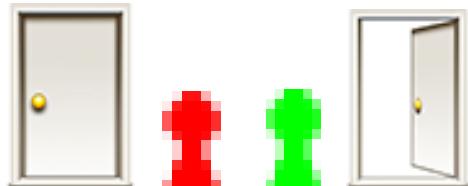
### Dispozitive IoT conectate la tabletă

**Smart Sprinkler** – sistem de irigare



În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **smart sprinkler** pentru a porni sau opri apa.

**Smart Door** – ușă inteligentă → pentru siguranță



Ușa inteligentă este în prezent deschisă (reprezentată de o lumină verde pe mânerul ușii), dar poate fi închisă de la distanță. Dă clic pe ușă inteligentă în browser pentru a extinde opțiunile. Dă clic pe **Lock** pentru a închide ușa (reprezentată de o lumină roșie pe mânerul ușii). Dă clic pe **Unlock** pentru a deschide ușa.

<sup>4</sup> <https://www.youtube.com/watch?v=9drBPsHAAww&t=326s>

În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **ușă** pentru a o deschide sau închide.

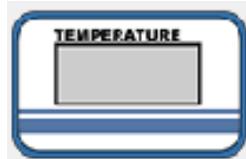
### Smart Fan – ventilator intelligent



Ventilatorul este în prezent oprit (nu are nicio linie în jurul lui), dar poate fi pornit de la distanță. Dă clic pe **Fan** în browser pentru a extinde opțiunile. Dă clic pe **High** pentru a porni ventilatorul (liniile din jurul ventilatorului). Dă clic pe **low** pentru a scădea intensitatea. Dă clic pe **off** pentru a opri.

În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **fan** pentru a porni sau opri ventilatorul.

### Temperature Meter – senzor de temperatură



### Smart Lamp – lampă intelligentă



Lampa este în prezent oprită, dar poate fi pornită de la distanță. Dă clic pe **Lampă** în browser pentru a extinde opțiunile. Dă clic pe **On** pentru a porni lampa (se aprinde un cerc colorat cu galben deschis). Dă clic pe **off** pentru a opri.

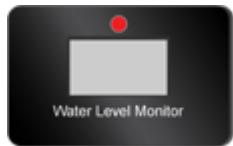
În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **lampa** pentru a porni sau opri lumina.

### **Smart Coffee Maker** – cafetieră inteligentă



În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **Coffee Maker** pentru a porni sau opri cafetiera inteligenta. Când este pornită avem în stânga acel buton roșu.

### **Smart Water Meter** – senzor nivel apă



### **Smoke Detector** – detector de fum



În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **Smoke Detcor** pentru a porni sau opri.

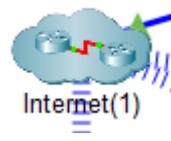
### **Smart Window** – fereastră inteligentă



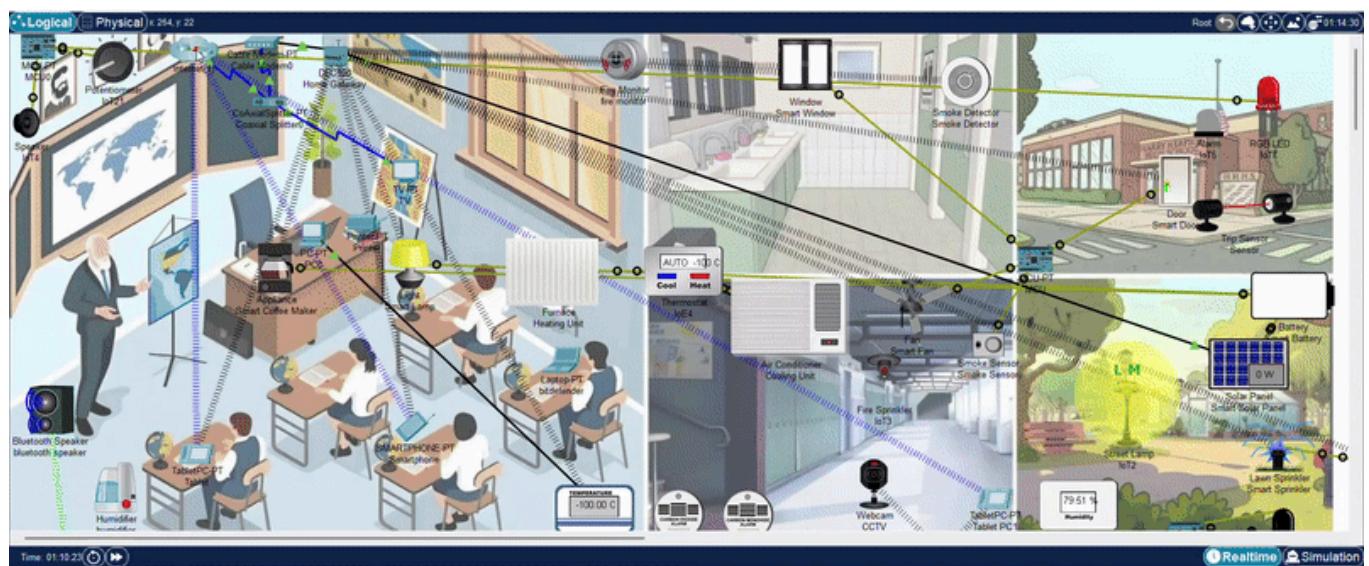
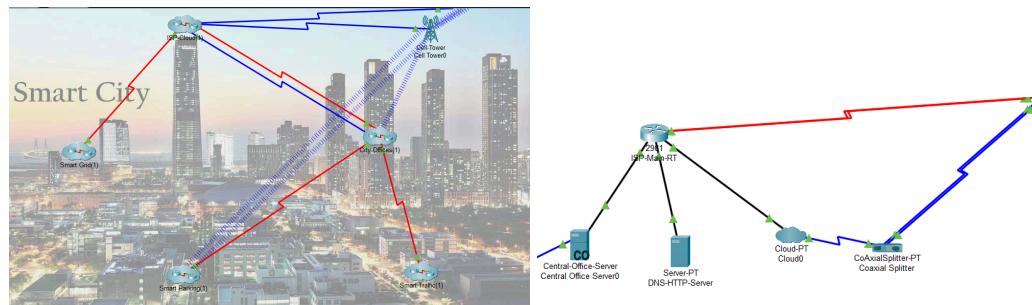
Fereastra este în prezent oprită, dar poate fi pornită de la distanță. Dă clic pe **Window** în browser pentru a extinde opțiunile. Dă clic pe **On** pentru a deshida fereastra (geamurile se vor deschide). Dă clic pe **off** pentru a închide.

În zona **Logical** din Packet Tracer, ține apăsat **ALT** și dă clic pe **Window** pentru a deschide sau închide.

## Obiectele din rețeaua principală



**Internet (Cloud)** – sursa principală de acces la internet



**Cable Modem** – conectează rețeaua locală la internet

**Coaxial Splitter** – împarte conexiunea de internet pentru dispozitive

**TV-PT** – televizor conectat la internet

**SMARTPHONE-PT** – smartphone conectat la internet

**DLC100 Home Gateway** – router IoT care gestionează dispozitivele inteligente

**TabletPC-PT** – tabletă care controlează dispozitivele IoT și are acces la internet

## Conecțarea:

Toate dispozitivele IoT sunt conectate prin Wi-Fi sau Zigbee la Home Gateway (DLC100). Tableta (TabletPC-PT) accesează Home Gateway și controlează toate dispozitivele inteligente prin intermediul aplicației „IoT Monitor”.

Rețeaua principală de internet: Cable Modem primește internet de la Cloud. Coaxial Splitter împarte semnalul către: TV-PT, SMARTPHONE-PT, DLC100 Home Gateway (care, la rândul său, controlează dispozitivele IoT). **TabletPC-PT** este conectată la internet prin **Wi-Fi**, dar și la **Home Gateway**, astfel încât poate **controla atât dispozitive IoT**, cât și **accesa internetul**.

## Interdependența între dispozitive și componente

- Sistemul este construit astfel încât dispozitivele IoT sunt dependente de **Home Gateway** pentru a putea fi controlate.
- **Tableta (TabletPC-PT)** este **interfața de control**, iar **Home Gateway (DLC100)** **gestionează comunicația între dispozitive**.

Internetul este distribuit prin Cable Modem → Coaxial Splitter → Home Gateway, asigurând conectivitatea necesară. Toate dispozitivele răspund corect la comenzi, iar utilizatorul poate controla sistemul de pe tabletă.

## Blinking an **LED / RGB LED** Using Blockly

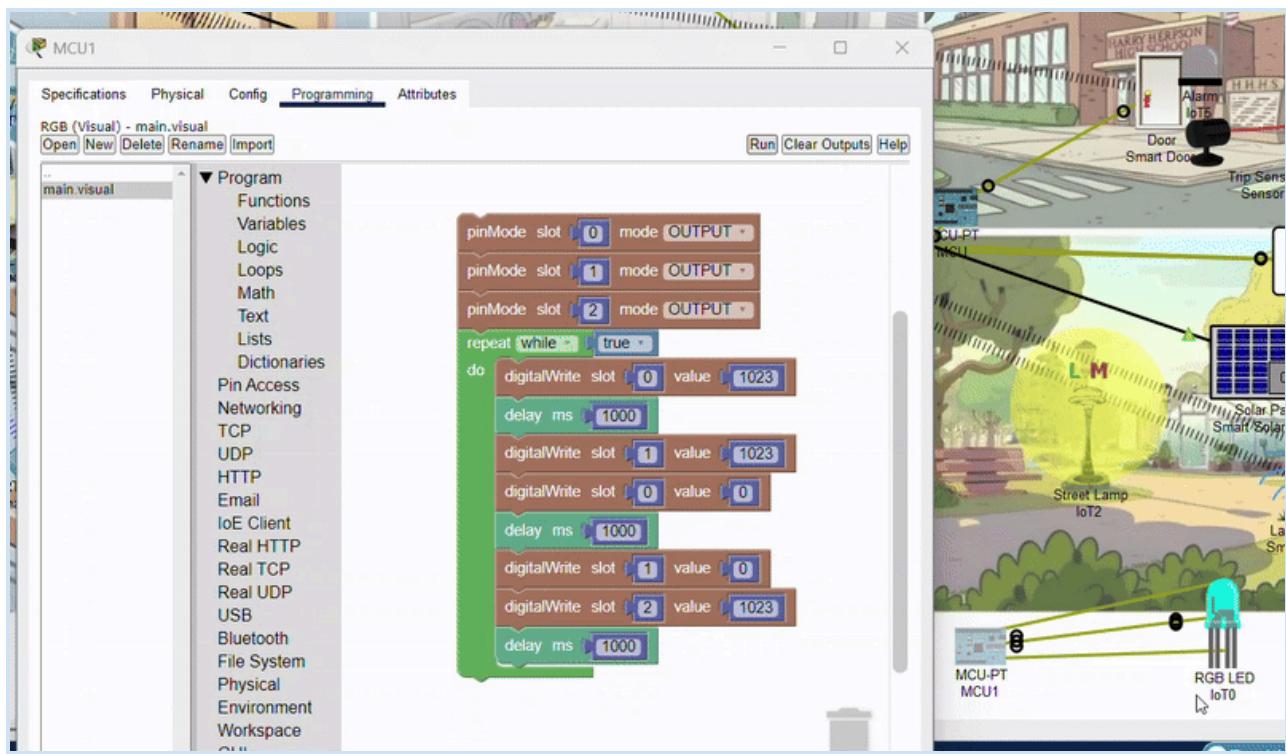
<sup>5</sup>**Utilizare:** ore de fizică, experimente, activități extrașcolare

Conecțate prin Costum Cable  (D0 - A0, D1-A1, D2-A2)

Apoi partea de **programabilitate** din clipul atașat mai jos, pentru **schimbarea succesivă a culorilor**.

---

<sup>5</sup> <https://www.youtube.com/watch?v=eYqe9up7saM&t=158s>



```

pinMode slot 0 mode OUTPUT
pinMode slot 1 mode OUTPUT
pinMode slot 2 mode OUTPUT

repeat while true do
    digitalWrite slot 0 value 1023
    delay ms 1000
    digitalWrite slot 1 value 1023
    digitalWrite slot 0 value 0
    delay ms 1000
    digitalWrite slot 1 value 0
    digitalWrite slot 2 value 1023
    delay ms 1000

```

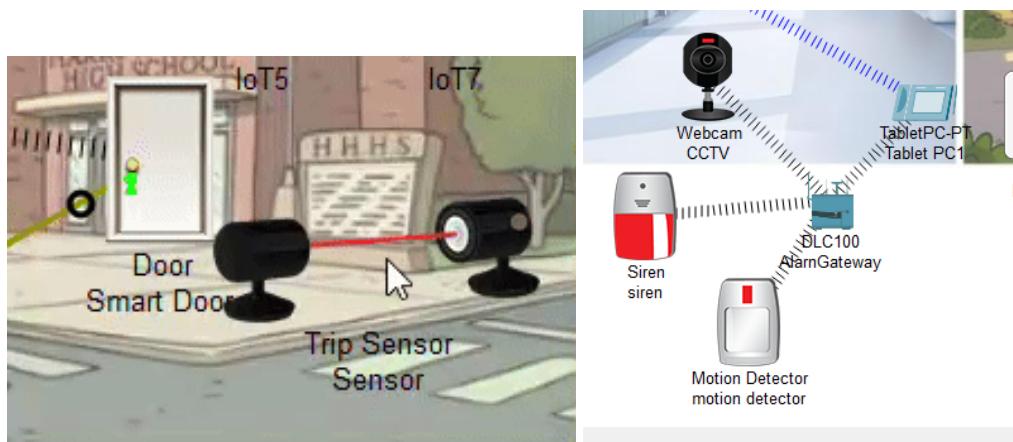
*pinMode slot 0 mode OUTPUT  
 pinMode slot 1 mode OUTPUT  
 pinMode slot 2 mode OUTPUT*

*repeat while true do  
 digitalWrite slot 0 value 1023  
 delay ms 1000*

*digitalWrite slot 1 value 1023  
 digitalWrite slot 0 value 0  
 delay ms 1000*

*digitalWrite slot 2 value 1023  
 digitalWrite slot 1 value 0  
 delay ms*

## Senzori



## Condițiile pe baza cărora funcționează sistemul de securitate

Actions	Enabled	Name	Condition	Actions
Edit Remove	Yes	MOTIONDETECTOR	PTT0810VL9R- On is true	Set PTT0810M7HZ- On to 1 Set PTT0810ZJ4T- On to 1
Edit Remove	Yes	NOMOTION	PTT0810VL9R- On is false	Set PTT0810M7HZ- On to 0 Set PTT0810ZJ4T- On to 0

## Experiența noastră cu simulatorul Cisco Packet Tracer

(din formularul pentru proiect)

Experiența noastră cu simulatorul Cisco Packet Tracer a fost una foarte practică și educativă. Proiectul SmartSchool ne-a oferit ocazia să aplicăm concepțele învățate despre rețele, dispozitive inteligente și automatizări într-un scenariu real și util – o școală intelligentă. Am învățat cum să configuroram o rețea funcțională, să conectăm dispozitive smart (precum senzori de fum, camere, uși inteligente, sisteme de iluminat etc.) și să le controlăm printr-un gateway central. Nu ne aşteptam ca aplicația să aibă atât de multe funcționalități interesante și să meargă totul atât de bine (se schimbă din zi în noapte, fiecare obiect are funcționalități, explicații, moduri diferite de conectare și programare).

Ce ni s-a părut dificil:

În primul rând, din cauză că probabil aplicația este destul de veche, am avut probleme cu aceasta, deoarece mergea foarte greu, se oprea, se închidea brusc și nu se ștergea tot progresul. Chiar de mai multe ori a trebuit să luăm totul de la capăt din cauză că aplicația se închidea și proiectul era șters. Însă, atunci când am reușit să trecem peste aceste probleme, a fost în

regulă. A fost bine că ne-am putut ajuta de tutoriale care explicau pas cu pas foarte bine unele conexiuni și cum să interacționăm cu obiectele.

Printre cele mai interesante lucruri pe care le-am făcut au fost:

înțelegerea modului în care comunica dispozitivele între ele (rețea locală vs cloud), conectarea corectă a dispozitivelor smart la Home Gateway și setarea interfeței web, simularea acțiunilor automate (precum deschiderea ușii sau pornirea ventilatorului în funcție de un senzor).

Prin acest proiect, am dobândit abilități utile pentru carierele noastre în IT: găndire logică, planificare de rețea, noțiuni de Internet of Things (IoT) și simulare a funcționalității dispozitivelor inteligente.

Cum ne va ajuta în carieră și la CV:

Acet proiect poate fi menționat la secțiunea "Proiecte" din CV, ca exemplu concret de lucru cu tehnologii smart și rețelistică de bază. Demonstrează că putem lucra într-un mediu de simulare, că înțeleg cum funcționează o rețea intelligentă și că pot contribui la soluții moderne în educație sau alte domenii prin tehnologie. De asemenea considerăm că IoT este un domeniu foarte de viitor cu care o să ne putem mândri, în special și pentru că acum știm să utilizăm aplicația Cisco Packet Tracer.

## BIBLIOGRAFIE

1. SMART HOME IOT PROJECT with cisco packet tracer stimulation tool || Packet tracer CISCO → ne-am inspirat la conectarea obiectelor cu tableta și smartphone  
<https://www.youtube.com/watch?v=7v1aAfn6Lww>
2. Introduction to IoT | Connecting & Simulating IoT Devices in Packet Tracer | IoT Registration Server  
<https://www.youtube.com/watch?v=EpCb7JiWIpo&t=226s>  
Intruder Detector or Thief-Catcher IoT System with Motion Detector, Siren & CCTV Using Packet Tracer → pentru a conecta obiectele  
[https://www.youtube.com/watch?v=8C1a3pK\\_Lpc&t=505s](https://www.youtube.com/watch?v=8C1a3pK_Lpc&t=505s)
3. Smart DAY/NIGHT IoT System Simulation with Smart Lamp, Ceiling Fan, Door & Windows in Packet Tracer  
<https://www.youtube.com/watch?v=RKbwkzfCn7k&t=1s>
4. Design and Simulation of IoT Systems Using the Cisco Packet Tracer | Smart Home IoT projects in PKT  
<https://www.youtube.com/watch?v=qbh7SzCfM1o&t=3s>
5. Static Routing in Cisco Packet Tracer. Static Routing in Hindi.
6. <https://www.youtube.com/watch?v=wErIwdU-vQA>
7. DHCP Through Static Routing In Cisco Packet Tracer. Networking In Hindi.  
<https://www.youtube.com/watch?v=unDgMK8V5HQ>
8. Cisco Packet Tracer : Blinking an LED / RGB LED Using Blockly  
<https://www.youtube.com/watch?v=eYqe9up7saM>
9. Carbon Monoxide Detector | Cisco Packet Tracer | IOT | Part 3  
<https://www.youtube.com/watch?v=CNPiu7cfPDc>
10. Cisco Packet Tracer - Listening to Bluetooth Music  
<https://www.youtube.com/watch?v=ucs0DWLSdjo&t=116s>
11. Bluetooth in Cisco Packet Tracer | Bluetooth Configuration → pentru conectarea Music Player cu boxa  
<https://www.youtube.com/watch?v=KFOjxa5RCPo>
12. #iot project #air conditioner #temperature control,#humidity monitor through smart phone iot project → am văzut cum s-a făcut conectarea între termostat, calorifer și aer condiționat  
<https://www.youtube.com/watch?v=-nD7jKrsu0o&t=167s>
13. Explore the smart home → ne-am inspirat pentru conectarea tabletei la obiectele din școală, pentru rețeaua de internet, ca să ne putem conecta și prin IoT Monitor dar și prin Web Browser  
<https://www.youtube.com/watch?v=9drBPsHAAww&t=324s>