

Requirements Documentation - Devices

HomeDork - Interactive House

Project Members

Reference	Name	Email
A	Samuel McMurray	Samuel_joseph.mcmurray0004@stud.hkr.se
B	Mustafa Ismail	mustafa.ismail0007@stud.hkr.se
C	Ibrahim Ahmed Ali	ibrahim.ahmed_ali0003@stud.hkr.se
D	Osayomore Edugie	Osayomore.edugie0004@stud.hkr.se

Revision History

Date	Version	Description	Author
18/9/2021	1.0	Initial requirements	A, B, C
07/10/2021	1.1	Added – Arm/Disarm Alarm	A, B, C, D
13/10/2021	1.2	Updated – The priority of the requirements Added – Read Temperature Outdoors, Radiator Temperature Adjustment, Timers, Stove On/Off, Window Open/Close, Water Leakage, Power Cut Off, Automatic Fire Alarm, Twilight Automatic System	A, B, C, D

Requirements List

Requirement Name	Priority
R1. Start/Run Device System	Essential
R2. Receive Commands	Essential
R3. Turn Lights ON/OFF	Essential

R4. Turn ON/OFF Fan	Essential
R5. Read Temperature Indoors	Essential
R6. Read temperature Outdoors	Essential
R7. Adjust Temperature Indoors	Essential
R8. Radiator Temperature Adjustment	Essential
R9. Timers	Essential
R10. Stove On/Off	Essential
R11. Window Open/Close	Essential
R12. Water Leakage	Essential
R13. Power Cut Off	Essential
R14. Arm Security Alarm	Essential
R15. Disarm Security Alarm	Essential
R16. Automatic Fire Alarm	Essential
R17. Twilight Automatic System	Essential
R18. Bluetooth Connection of Devices	Desirable
R19. Turn ON/OFF Fan Oscillation	Desirable
R20. Set Fan Timer	Desirable
R21. Adjust curtains	Desirable
R22. Some Lights on Dimmers	Desirable
R23. Adjust Fan Speed	Desirable

Requirements Descriptions

R1

The Arduino should run with out major errors or crashing requiring the system to reboot.

R2

The Arduino shall be able to receive commands from an external server then execute those commands locally activating or deactivating a device.

R3

The Arduino device shall be able to turn on and off specific lights based off the commands received from the server. The smart house model currently has an indoor and outdoor light.

R4

The Arduino device shall be able to turn on and off specific fans based off the commands received from the server. The smart house model currently has 1 fan located in the loft.

R5

The Arduino device shall be able to get a temperature reading of the house indoors with the connected thermometer and relay that temperature to the server. The smart house model contains thermometers located on the inside of the house which use the LM35C temperature sensor.

R6

The Arduino device shall be able to get a temperature reading of the house outdoors with the connected thermometer and relay that temperature to the server. The smart house has an external temperature sensor that uses the SMT160-30.

R7

The Arduino device shall be able to adjust the temperature in the house to reflect the desired temperature received from the server by adjusting a fan or a radiator. The smart house model contains a fan located in the loft which could be activated with the intention to cool the house, the model also contains radiators which will be adjusted to heat the house.

R8

The Arduino device shall be able to adjust the temperature radiator based off the desired temperature in the home. The smart house contains 4 radiators which use voltage between the range of 0 and 5 to adjust the output strength with 5V being the highest temperature setting.

R9

The Arduino device shall be able to start a timer that the user can determine through a command from the server. The timers will turn on a light when the timer is up.

R10

The Arduino device shall be able to turn on and off a stove through commands received from the server. The smart house model contains a switch that will simulate the signal for this operation.

R11

The Arduino device shall be able to open and close a window based off the commands received from the server. The smart house model contains a switch that will simulate the signal for this operation.

R12

The Arduino device shall be able to detect water leakage and relay that information to the end-user. The smart house contains a switch that will simulate the signal for this operation.

R13

The Arduino device shall be able to detect a power outage, currently the power outage will be simulated by the 12V supply voltage.

R14

The Arduino device shall be able to arm the alarm based off commands received from the server. The alarm while active will use a magnetic sensor located at the front door of the smart house model to demonstrate the break in alarm function. The alarm will have 2 components a light located at the top of the house near the chimney and a loudspeaker siren that will sound.

R15

The Arduino device shall be able to disarm the alarm based off commands received from the server. The security alarm while disarmed will not activate when the door is open.

R16

The Arduino device shall be able to run a fire alarm independent of the other devices. The fire alarm will be tested using a switch to simulate a sensor for the alarm being triggered.

R17

The Arduino device shall have a twilight automatic system that runs on the outdoor lights once the sun begins to set and turns off once the sun rises. The twilight automatic system uses the TSL250 sensor to measure the brightness.

R18

The Arduino device should be able to connect to and issue commands received from the server to the devices in the house using Bluetooth.

R19

The Arduino device should be able to turn on and off oscillation of a specific fan based off the commands received from the server.

R20

The Arduino device should be able to set a timer on a specific fan based off the commands received from the server.

R21

The Arduino device should be able to adjust the curtains state of up or down based on the commands received from the server.

R22

The Arduino device should have the ability to dim certain lights based off the percentage of dim that is received from the server.

R23

The Arduino device should be able to adjust the specific fans speed based off the commands received from the server. The smart house model contains a single fan in the loft which uses PWM so this can be done using an integer between 0 and 255 0 being off and 255 being the top speed.