Touch Less Elevator Panel Design

Aditiya Ranganath Rao (61570)
Embedded systems and electronics
Chennai, India
ISEP Paris, France
aditya.ranganath-rao@eleve.isep.fr

Joseph Nirmalraj Jeyanathan (61572)
Embedded systems and electronics
Madurai, India
ISEP Paris, France
Joseph.jeyanathan@elevel.isep.fr

Walid.naouss (61678)
Embedded systems and electronics
Akkar, Lebanon
ISEP Paris, France
wana61678@eleve.isep.fr

Wu Keyang (61558)
Embedded systems and electronics
Langfang, China
ISEP Paris, France
kewu61558@eleve.isep.fr

Ye Hang (61557)
Embedded systems and electronics
Kunming, China
ISEP Paris, France
haye61557@eleve.isep.fr

Abstract **—** The embedded electronic engineering project was a group 1 project to build a touch less elevator panel in team of five members. The Touch Less Elevator panel had been designed to relating to a device that begins to operate when it senses an object or presence, without needed to be touched. Each member of the team had specific topic to concentrate on. Consequently, this report details the development of the touch less panel design, administrative report, software and hardware integration, testing and mentions the project management activities. The object identification and audio output is the top-level aspect of the panel. The tasks it must accomplish can be roughly broken down into four main components: Identification of an object, Produce output with an audio format, memory location of audio input and utilization of aurdino controller.

I. PROJECT SUMMARY

A. INTRODUCTION

The project is entitled the 'Touch less elevator panel' (TLEP). The touch less elevator panels had been designed to relating to a device that begins to operate when it senses an object or presence, without needing to be touched, it is specially designed for helping in pandemic situation. As per Research, There is 3500 Bacteria's Per Square Inch on Lift Button. Lift Button has 17 Times Greater Bacteria than the Toilet Seat. And in current Scenario of COVID-19, there are huge chances that someone gets infected because of using Elevator Buttons. This interface would give less chances of getting infected by using of elevator buttons.

The project description was to build a touch less panel in a team of five. Although this is a group project, the assessment is individual. Hence each member of the team was set specific tasks that will count as an individual project, but eventually would join together to make a touch less operation. One of the advantages of this group approach is that a meaningful project can be attempted, lots of different ideas implementation, platform for the collaboration, more progressive, which can be continued.

This report details the development of the touch less elevator panel and mentions the project management. The object identification and audio output is the top-level aspect of the panel. The tasks it must accomplish can be roughly broken down into four main components: Identification of an object, Produce output with an audio format, memory location of audio input and utilization of aurdino controller. There are sub tasks for all the main components like Selection of sensor, controller kit, extensive thought in approaching with outputs, power calculation, sensor specification fitness, identifying memory space for an audio storage and outputs.

B. *Project goals and objectives*

The main goal and objective of this project is to implement touch less elevator panel for upcoming installation lifts for commercial and industries to prevent touching the lift buttons due to ease chance of infecting by viruses.

- Contact less action
- Ease of use
- GUI design
- Safety environment options

C. *Project Scope*

The project will introduce touch less elevator panel including the following features,

- Outfit Panel with active state LED identification
- Audio output
- Alarm state identification
- Software reset option

D. *Acknowledgement*

During this project period there are several ideas were discussed and implemented between team members for this project. Firstly we would like to thank Dr Xun Zhang (Isep) for providing this idea to us to do this project and for continuous support throughout the project. Secondly we would like to thank our all team members for very good collaboration, ideas contributions, been cooperative and made the team work so well. There have been no occasions where a conflict of opinion has not been resolved successfully.

E. Responsibilities and Resources

Responsibilities &Owner						
Slno	**Tasks**	**Aditiya**	**Keyang**	**Walid**	**Hang**	**Joseph**
1	Project Plan	√	√	√	√	√
2	Risk and Mitigate actions	√	√	√	√	√
3	Identification of components	√	√	√	√	√
4	Procurement	√		√		
5	Components testing	√				√
6	Audio file conversion		√		√	
7	3D model design		√		√	
8	Hardware integration	√		√		√
9	Firmware development	√				√
10	Application development	√			√	√
11	Software and Hardware integration	√				√
12	Testing & Deliverables	√	√	√	√	√

F. Abbreviations and Acronyms

Slno	Shortcuts	Abbreviations and Acronyms
1	IR	Infrared Sensors
2	OP amplifier	Operational amplifier
3	IDE	Integrated development environment
4	PWM	Pulse width modulation
5	I/O	Inputs outputs
6	EEPROM	Electrically Erasable Programmable read-only memory
7	RAM	Random access memory
8	SPI	Serial Peripheral Interface
9	WAV	Waveform Audio File format
10	VCC	Voltage Common collector
11	GND	Ground terminal
12	MISO	Master In Slave Out
13	MOSI	Master Out Slave In
14	Clck	Clock signal
15	CS	Chip Select
16	GUI	Graphical User Interface

G. *Identification of Components*

Below are the components are identified for project purpose,

1. Infra red sensors

2. Speaker module
3. Power amplifier
4. Aurdino board
5. SD card module

II. Technical Description of components

A. *Infra red sensors*

This sensor module has great adaptive capability of the ambient light, having a pair of infrared transmitter and the receiver tube.

The infrared emitting tube to emit a certain frequency, encounters an obstacle detection direction (reflecting surface), infrared reflected back to the receiver tube receiving, after a comparator circuit processing, the green LED lights up, while the signal output will output digital signal (a low-level signal), through the potentiometer knob to adjust the detection distance, the effective distance range 2 ~ 5cm working voltage of 3.3V-5V. The detection range of the sensor can be adjusted by the potentiometer, with little interference, easy to assemble, easy to use features, can be widely used robot obstacle avoidance, obstacle avoidance car assembly line count and black-and-white line tracking and many other occasions.

The module features a 3 wire interface with Vcc, GND and an OUTPUT pin on its tail. Upon hindrance/reflectance, the output pin gives out a digital signal (a low level signal). The onboard preset helps to fine tune the range of operation, effective distance range is 2cm to 5cm.

IR Infrared Sensor Features:

- Easy to assemble and use
- Onboard detection indication
- Effective distance range of 2cm to 5cm
- Preset knob to fine tune distance range

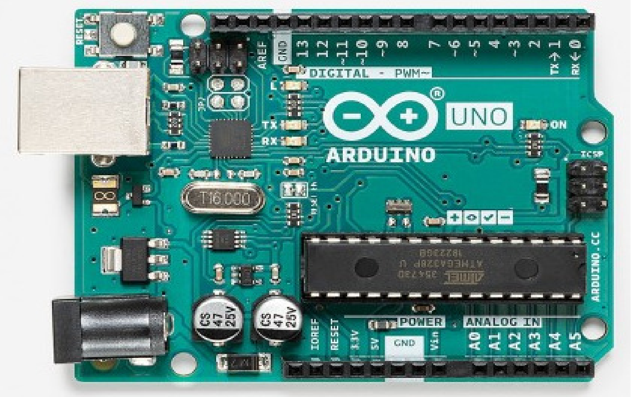
B. *Aurdino UNO board*

Arduino Uno is a microcontroller board based on the ATmega328P (datasheet). It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator (CSTCE16M0V53-R0), a USB connection, a power jack, an ICSP header and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started.

"Uno" means one in Italian and was chosen to mark the release of Arduino Software (IDE) 1.0. The Uno board and version 1.0 of Arduino Software (IDE) were the reference versions of Arduino, now evolved to newer releases. The Uno board is the first in a series of USB Arduino boards, and the reference model for the Arduino platform; for an extensive list of current, past or outdated boards see the Arduino index of boards.

Datasheet, http://ww1.microchip.com/downloads/en/DeviceDoc/doc7799.pdf
Main features: High performance, low power AVR® 8-bit microcontroller,

- 125 Powerful Instructions – Most Single Clock Cycle Execution
- 22 Programmable I/O Lines
- Operating Voltages – 2.7 - 5.5V
- 32 x 8 General Purpose Working Registers
- Fully Static Operation
- 8K/16K/32K Bytes of In-System Self-Programmable Flash
- 512/512/1024 EEPROM
- 512/512/1024 Internal SRAM
- Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
- USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion
- 4 Programmable Endpoints
- One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
- One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode (three 8-bit PWM channels)
- USART with SPI master only mode and hardware flow control (RTS/CTS)
- Master/Slave SPI Serial Interface



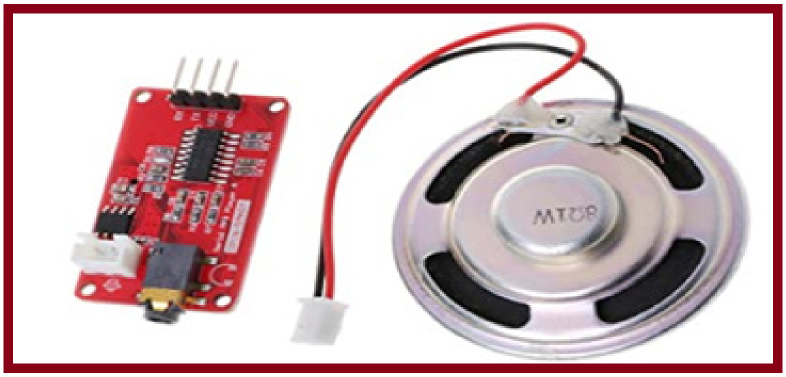
C. *Speaker module with power amplifier*

We were looking for the audio output for the project to make sure the user entered the correct button number when the panel button selection. Audio output over speaker required amplification for the tuned and good signal output.

The module is a kind of simple MP3 player which is based on high quality MP3 audio chip. It can support MP3 and WAV formats of 8k Hz ~ 48k Hz sample rate. There is a TF card socket on board, so that you can plug in the micro SD card which stores audio files. MCU can control MP3 playing state by sending commands to module through UART port, such as changing songs, changing volume and playing mode and so on. You can also debug the module via USB to the UART module. It is compatible with Arduino / AVR / ARM / PIC.

Characteristics:

- Support sampling frequency (kHz): 8 / 11.025 / 12/16 / 22.05 / 24/32 / 44.1 / 48
- Support file format: MP3 / WAV
- Support Micro SD card, Micro SDHC card
- Integrated 3 watt mono amplifier
- UART TTL serial control read mode, baud rate is 9600bps
- Serial communication format: 8N1
- Operating voltage: 3.7 ~ 5.25V DC
- On-board headphone jack, it allows you to connect headphones or an external amplifier.
- Integrated TF card connector
- Dimensions: 2.3x4.3cm
- 8ohm 1W speaker:
- Can be directly plugged into the serial MP3 player module to use.
- Magnetic forms: inner magnetic
- Rated impedance: 8 ohm
- Rated power: 0.5W
- Dimension: diameter 4.9cm / 1.93 inch
- Great for DIY speaker and amplifier projects



D. SD card module

SPI Reader mini memory SD TF memory card Board Module Shield for Arduino
Size: 4.1x 2.4cm Control

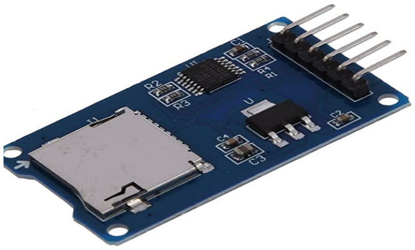
Interface:
A total of six pins (GND, VCC, MISO, MOSI, SCK, CS), GND to ground, VCC is l power, MISO, MOSI, SCK is the SPI bus, CS is the chip signal selector pin.

3.3V regulation circuit:
LDO regulator output level 3.3V conversion chip, mini SD card supply.

Level conversion circuit:
mini SD card in the direction of the signals in 3.3V is the miniSD card in the direction of the signal control interface MISO also converted 3.3V, can general AVR mini controller system to read the signal.

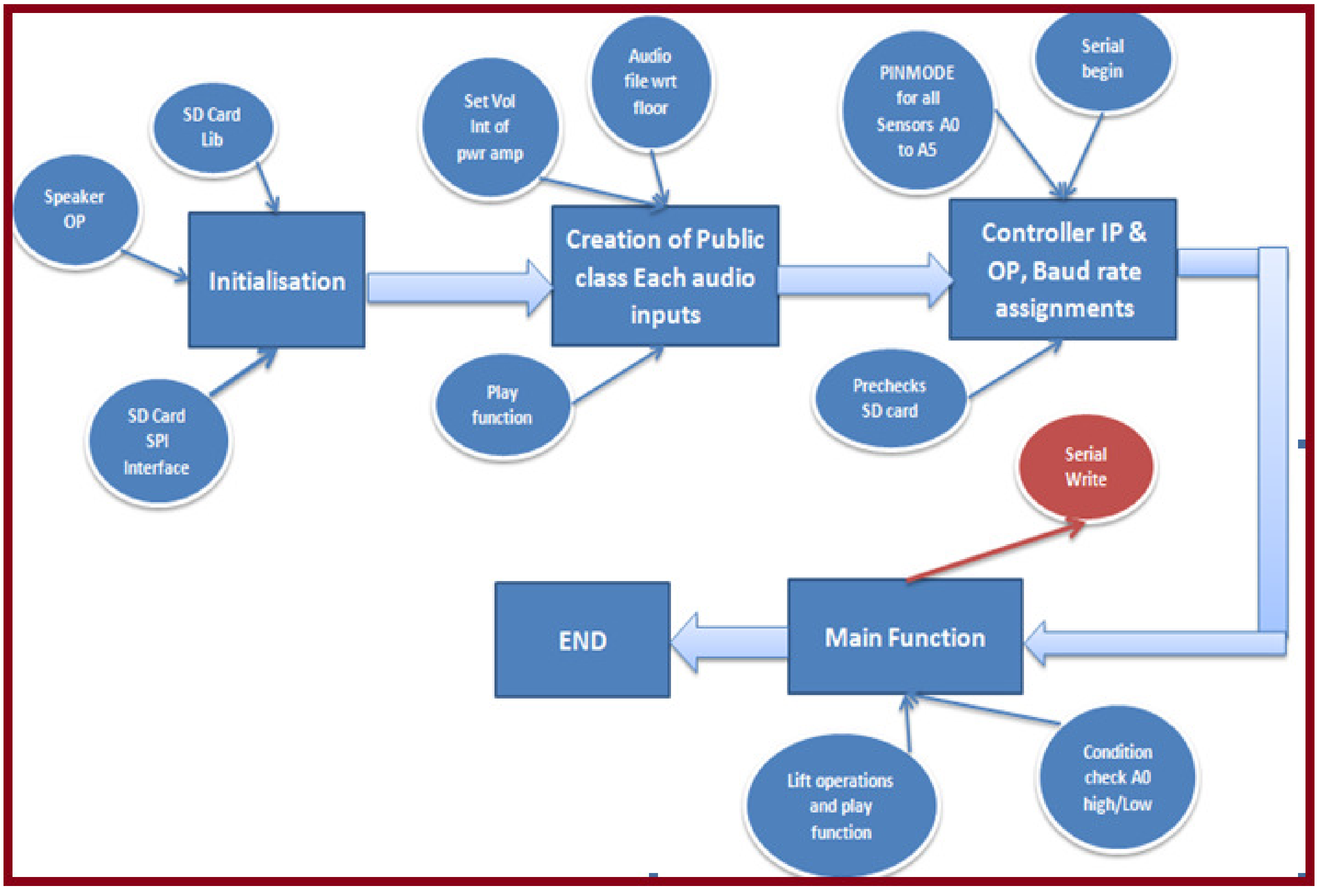
Mini SD card port:
From the bomb terrace for card insertion and easy removal.

Positioning holes:
4 M2 positioning screws 2.2mm hole diameter, the module is easy to install positioning in order to achieve the inter-module combination.



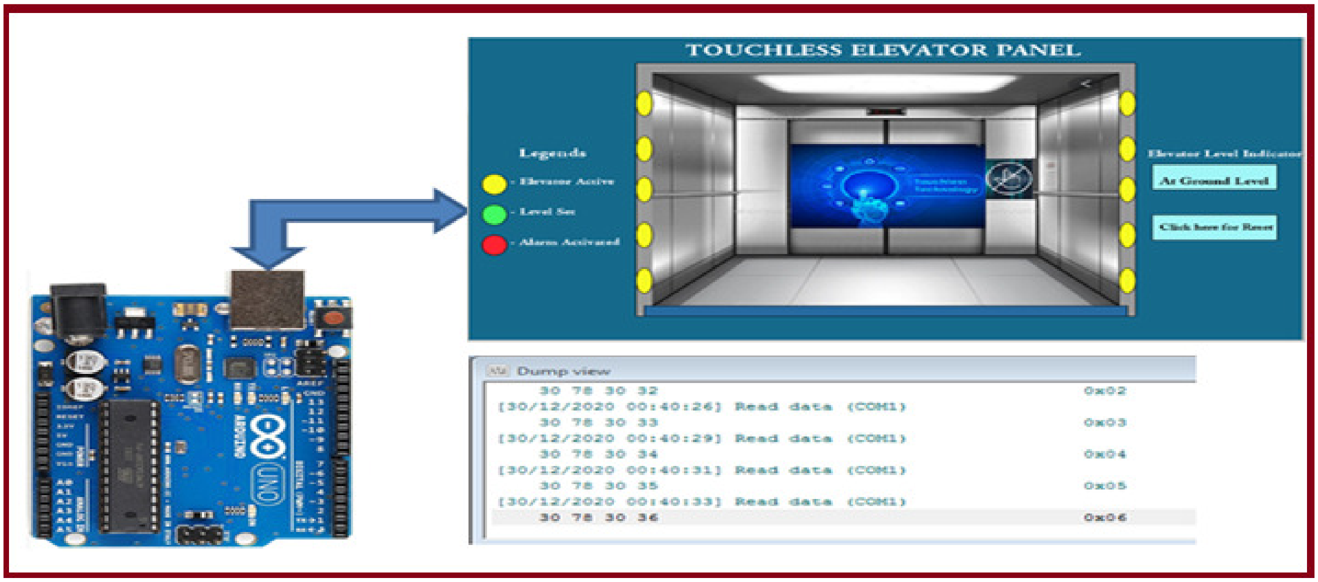
III. Firmware and software design

A. ***Function block diagram firmware*** - The firmware contains the main functionality for accessing to the hardware board with the integration of Sensors, Speaker, SD card with SPI communication and power amplifier. Below figure would help us to identify the firmware design flow for this implementation.

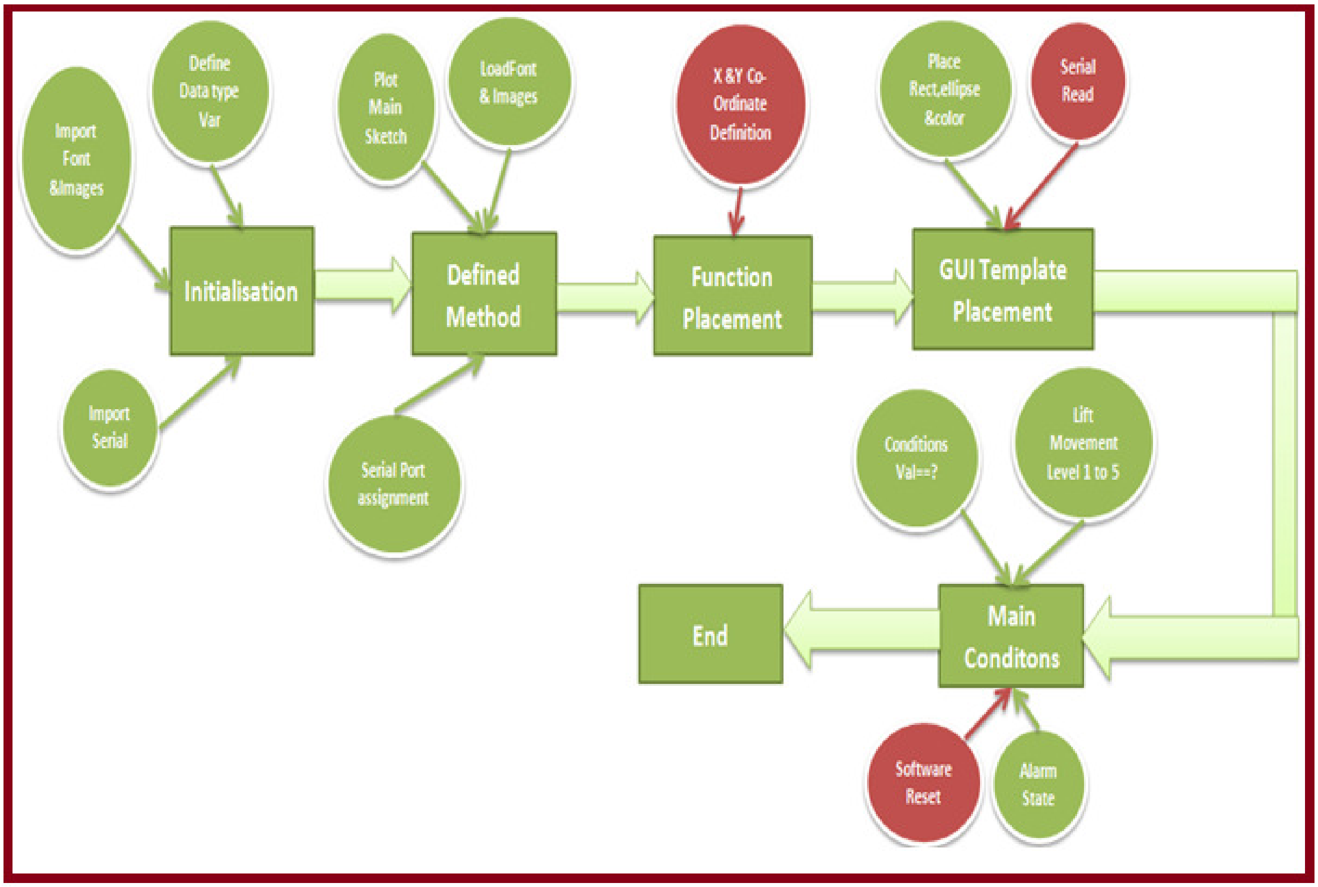

Speaker OP
SD Card Lib
Initialisation
SD Card SPI Interface
Set Vol Int of pwr amp
Audio file wrt floor
Creation of Public class Each audio inputs
Play function
PINMODE for all Sensors A0 to A5
Serial begin
Controller IP & OP, Baud rate assignments
Prechecks SD card
Serial Write
END
Main Function
Lift operations and play function
Condition check A0 high/Low

B. Function block diagram software application

The graphical user interface designed for accessing and monitoring the hardware board which was implemented above. We shall abele to monitor the power up condition, level of sensor states active or inactive or alarm and a movable object would help in the simulation view of elevator movement.



```
    30 78 30 32                                             0x02
[30/12/2020 00:40:26] Read data (COM1)
    30 78 30 33                                             0x03
[30/12/2020 00:40:29] Read data (COM1)
    30 78 30 34                                             0x04
[30/12/2020 00:40:31] Read data (COM1)
    30 78 30 35                                             0x05
[30/12/2020 00:40:33] Read data (COM1)
    30 78 30 36                                             0x06
```



IV. Process management

A. *Risk and Mitigate actions*

- Risk mitigation planning is the process of developing options and actions to enhance opportunities and reduce threats to project objectives.
- Risk mitigation implementation is the process of executing risk mitigation actions.
- Risk mitigation progress monitoring includes tracking identified risks, identifying new risks, and evaluating risk process effectiveness throughout the project.

Slno	Identified Risks	Risk Level	Mitigate Advice
1	Collision between two or more sensors when detects the obstacles	High	Install the IR sensor with No conflict. Installation should be perfect
2	Audio storage insufficient for storing the audio file output	Low	Utilize inbuilt SD card function of Aurdino UNO or Use external SD card and store the files
3	Memoring the previous value in EEPROM and set active and inactive state	Low	Check EEPROM memory of aurdino UNO to enhance this requirement, and consider the time of the project for deployment
4	Binary inputs passing by serial communication and GUI must understand the value	Medium	Use serial port monitor tools to identify the format of serial read and implement the values into the GUI conditions

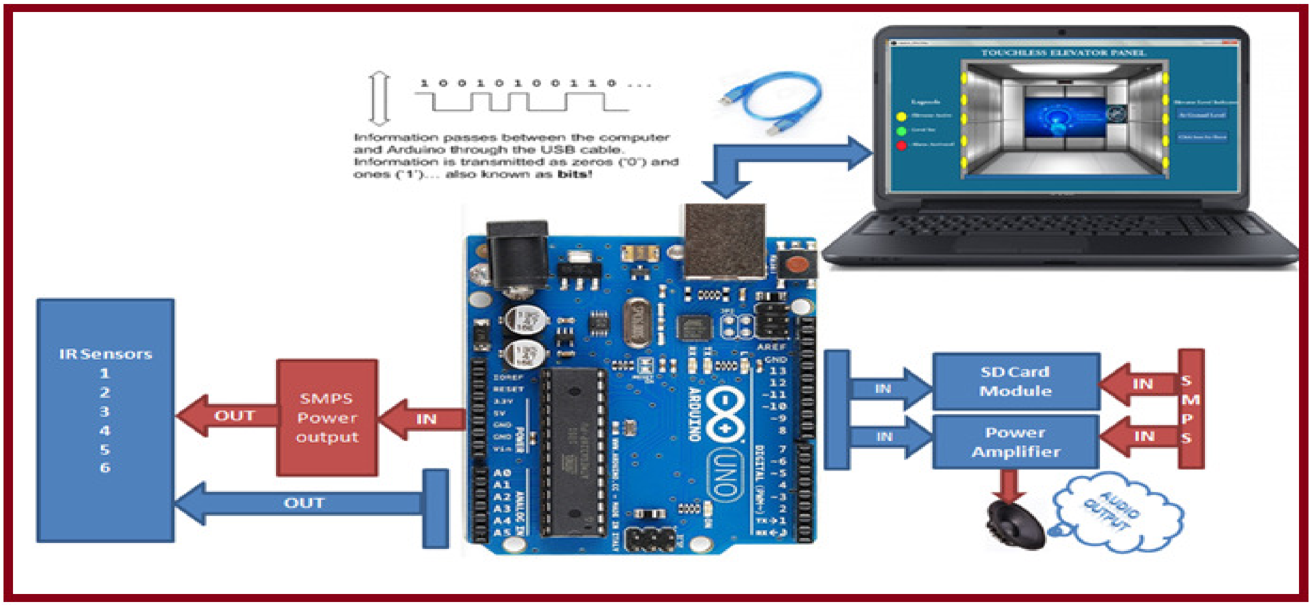
B. *Administrative plan and achieved state*

Slno	Components	Quantity	Price in euro
1	IR sensors	10	14.99
2	Speaker module	1	10.49
3	Op amplifier for speaker module	2	33.26
4	SD CARD module	1	11.99
5	Aurdino board	1	0
	Total		**70.73**

V. Project Result

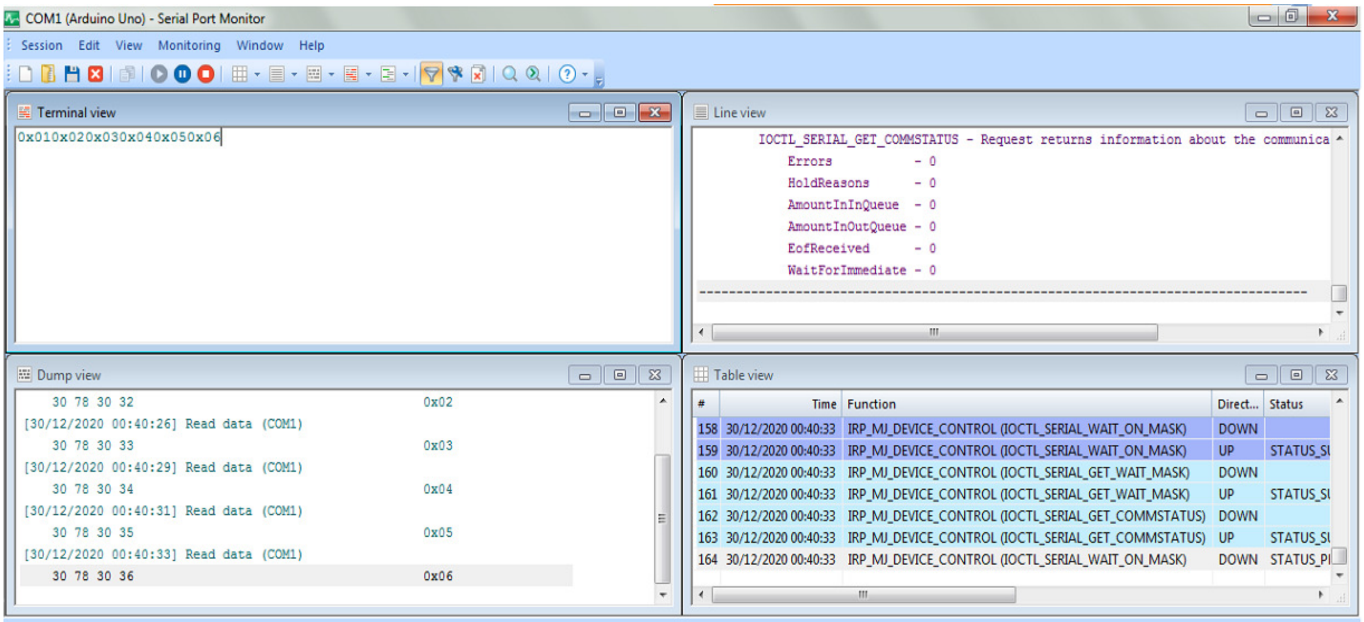
A. *Project Hardware and Software integration diagram*

Total project had been designed incorporated with few layers of OSI model, Sensor inputs are reading via serial communication and controller decides the working principle based on the firmware and transferring the data to GUI for the user visibility.



B. *Serial Port Monitor output*

It's a freeware tool helped for reading the serial data and converting into "processing" tool format.



```
0x010x020x030x040x050x06
```

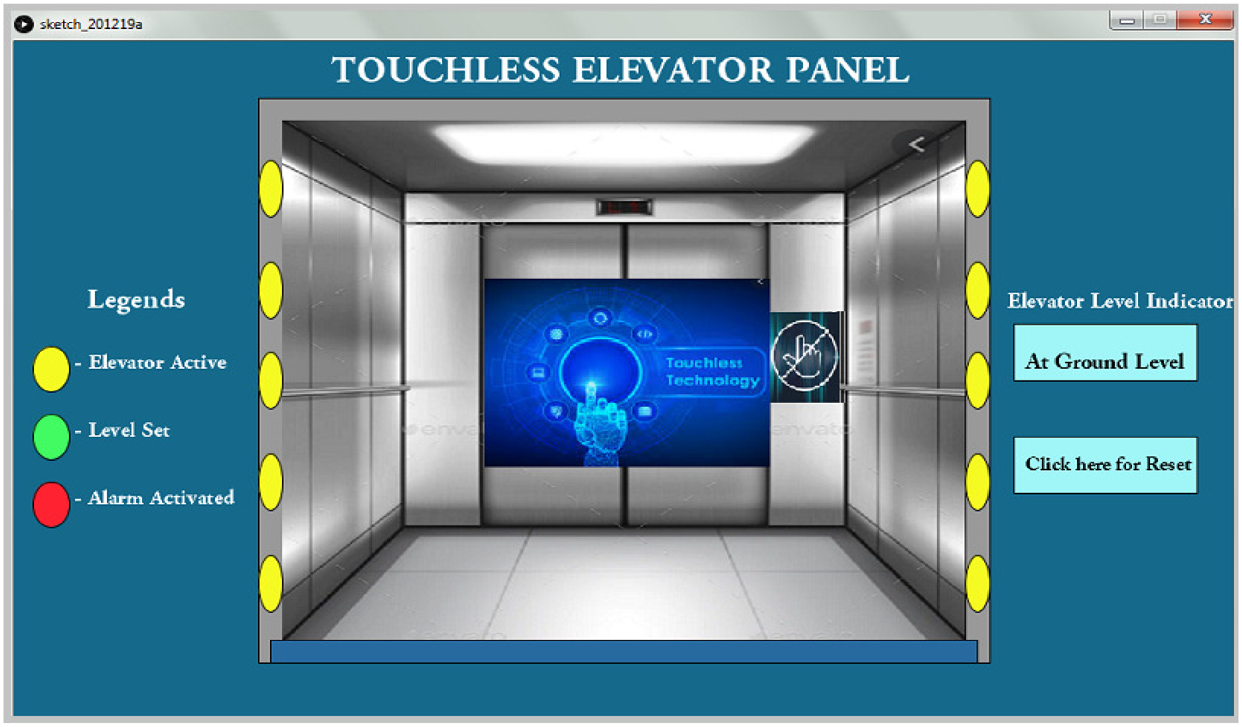
```
        IOCTL_SERIAL_GET_COMMSTATUS - Request returns information about the communica
            Errors           - 0
            HoldReasons      - 0
            AmountInInQueue  - 0
            AmountInOutQueue - 0
            EofReceived      - 0
            WaitForImmediate - 0
----------------------------------------------------------------------------------
```

```
    30 78 30 32                                            0x02
[30/12/2020 00:40:26] Read data (COM1)
    30 78 30 33                                            0x03
[30/12/2020 00:40:29] Read data (COM1)
    30 78 30 34                                            0x04
[30/12/2020 00:40:31] Read data (COM1)
    30 78 30 35                                            0x05
[30/12/2020 00:40:33] Read data (COM1)
    30 78 30 36                                            0x06
```

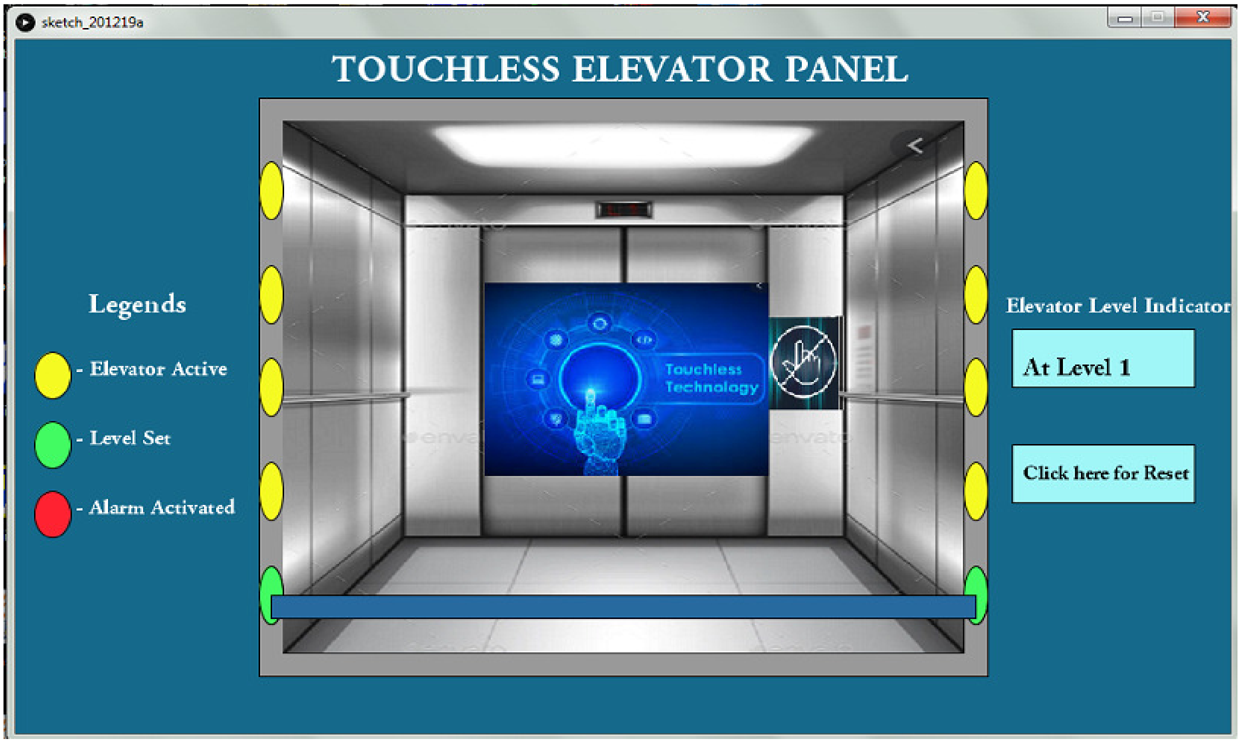
#	Time	Function	Direct...	Status
158	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_WAIT_ON_MASK)	DOWN	
159	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_WAIT_ON_MASK)	UP	STATUS_SU
160	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_WAIT_MASK)	DOWN	
161	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_WAIT_MASK)	UP	STATUS_SU
162	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_COMMSTATUS)	DOWN	
163	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_GET_COMMSTATUS)	UP	STATUS_SU
164	30/12/2020 00:40:33	IRP_MJ_DEVICE_CONTROL (IOCTL_SERIAL_WAIT_ON_MASK)	DOWN	STATUS_PI

C. GUI References for the Hardware output

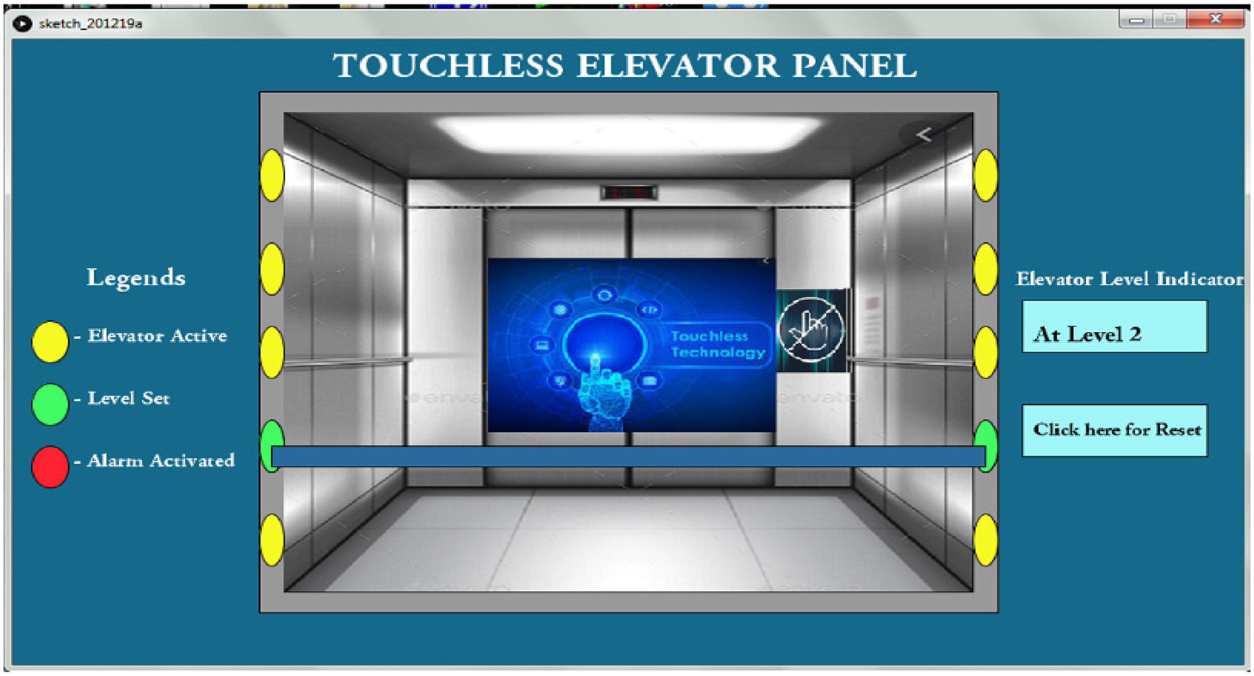
- Ground level indication and elevator active status (yellow_color indication)



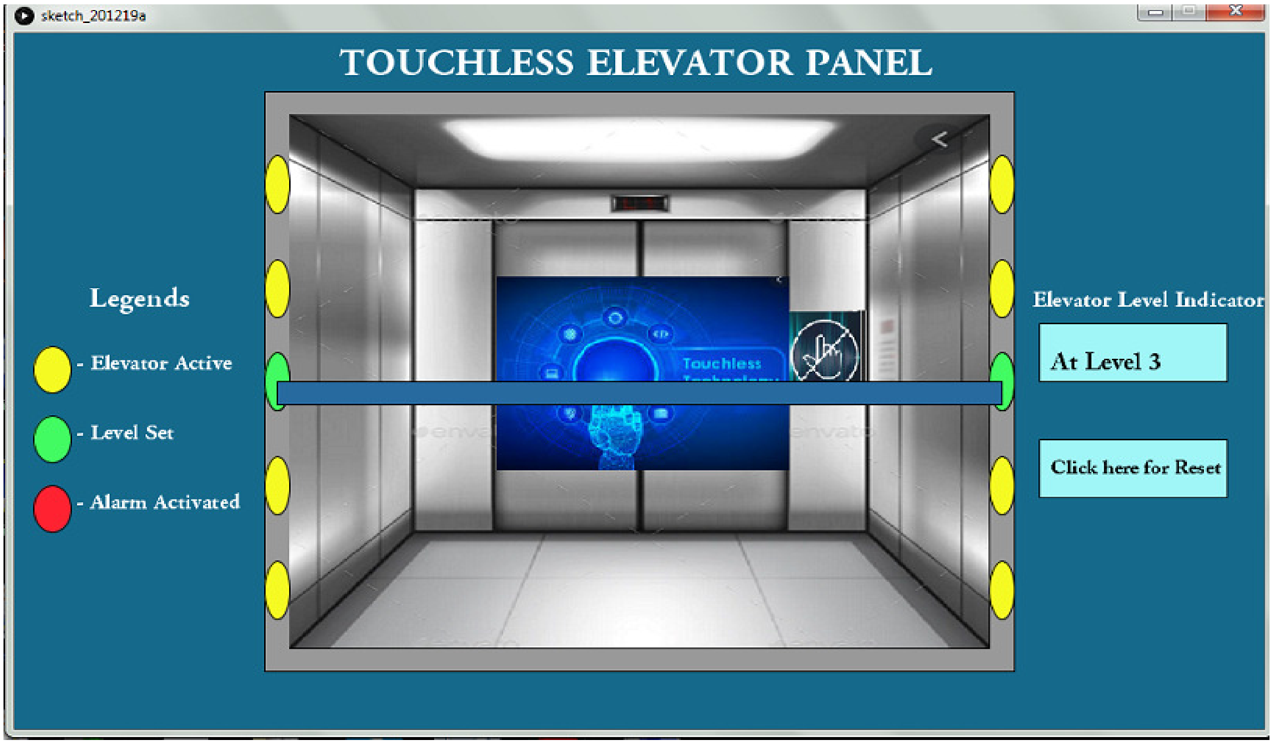
- Level 1 indication Sensor 1 active with Green color and moveable object reaches to level 1 and pending all sensors are status are active (yellow_color indication)



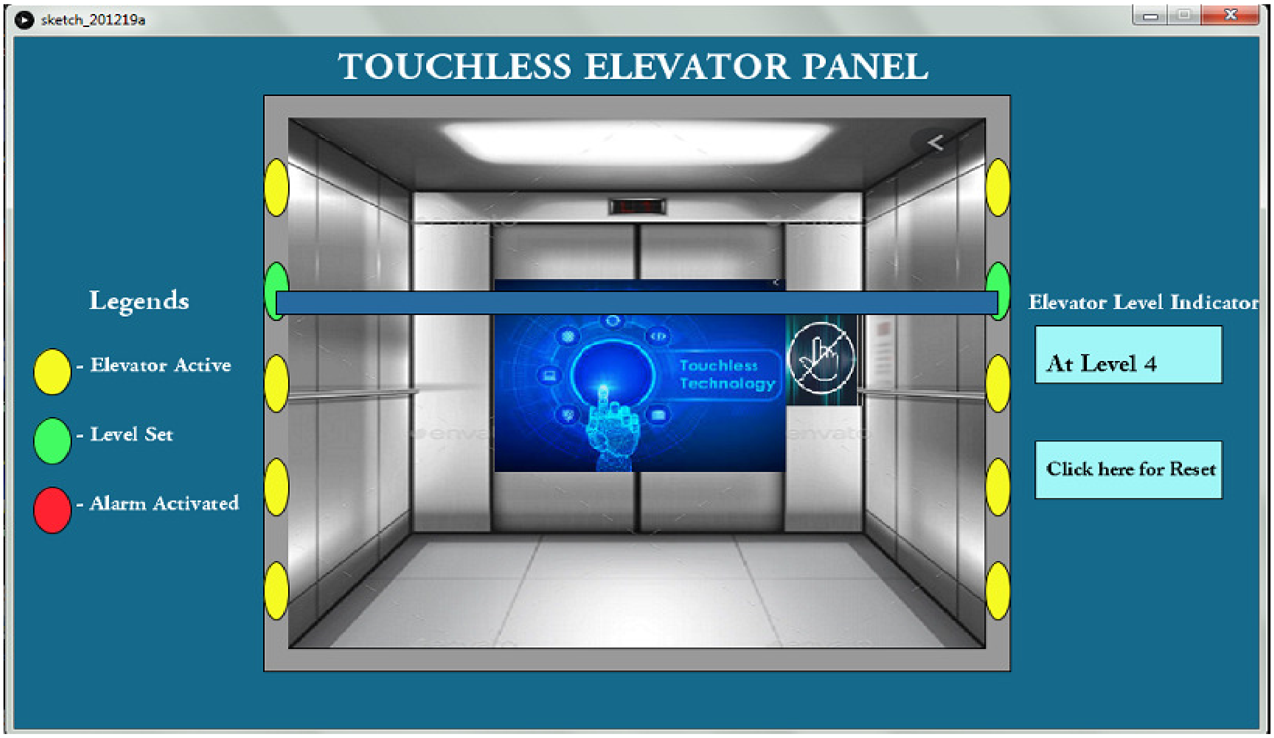
- Level 2 indication Sensor 2 active with Green color and moveable object reaches to level 2 and pending all sensors are status are active (yellow_color indication)



- Level 3 indication Sensor 3 active with Green color and moveable object reaches to level 3 and pending all sensors are status are active (yellow_color indication)



- Level 4 indication Sensor 4 active with Green color and moveable object reaches to level 4 and pending all sensors are status are active (yellow_color indication)



sketch_201219a

TOUCHLESS ELEVATOR PANEL

Legends

🟡 - Elevator Active

🟢 - Level Set

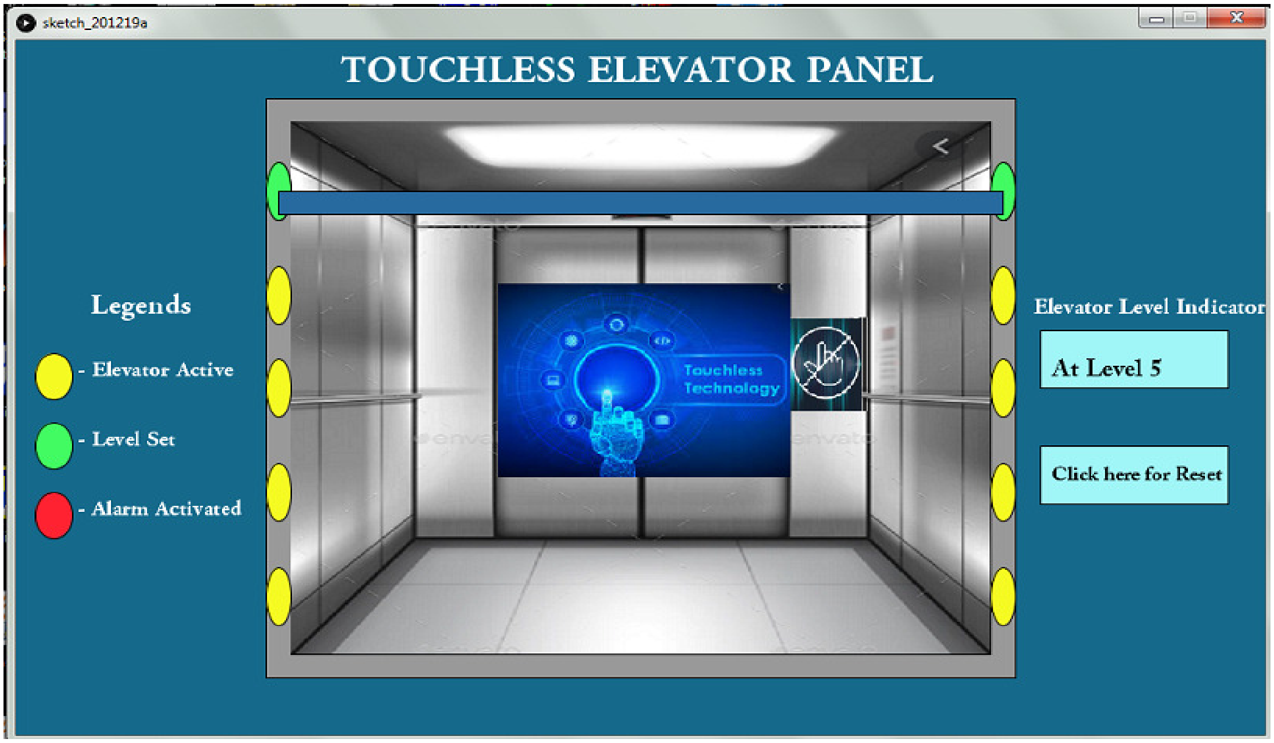
🔴 - Alarm Activated

Elevator Level Indicator

At Level 4

Click here for Reset

- Level 5 indication Sensor 5 active with Green color and moveable object reaches to level 5 and pending all sensors are status are active (yellow_color indication)



sketch_201219a

TOUCHLESS ELEVATOR PANEL

Legends

🟡 - Elevator Active

🟢 - Level Set

🔴 - Alarm Activated

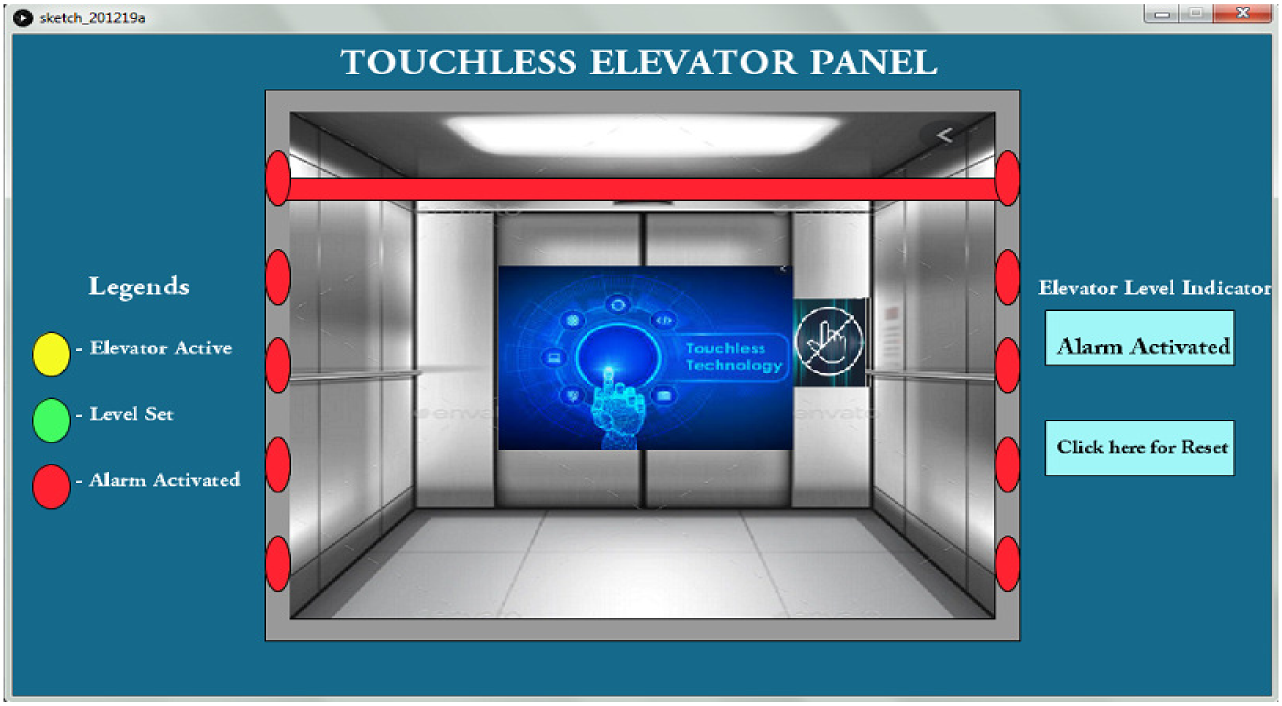
Elevator Level Indicator

At Level 5

Click here for Reset

Touch Less Elevator Panel design

- Alarm state active indication all sensor turns Red and moveable object stands or hold at the level where it was.
- Software reset shall make status normal or again activate any of the floor sensor for the normal initial state



D. Milestones (latency, power consumption, Problem statement & resolved state)

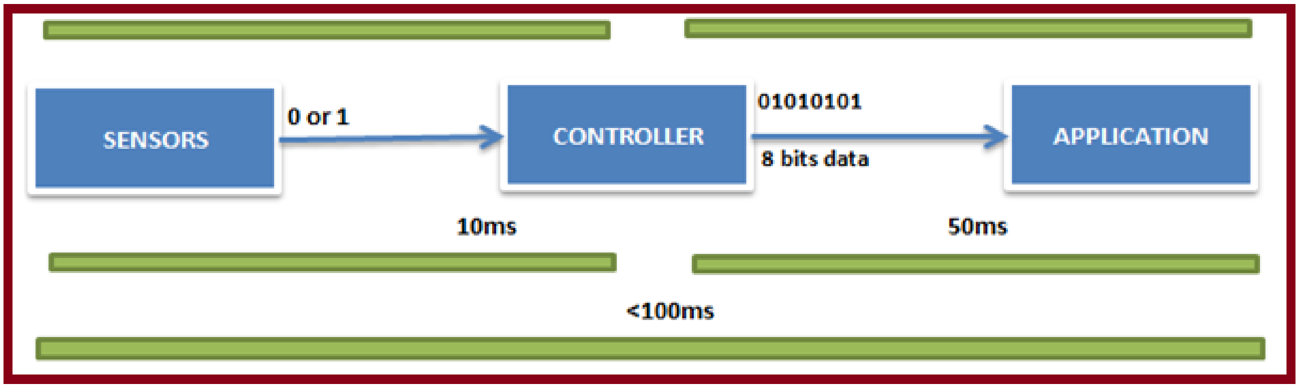
Latency - "Time delay between input event being applied to a system and the associated output action from the system"

For real-time embedded systems that process and respond to sensor data, a key measure of network performance is how long it takes data to get from one device to another – in other words, latency.

Example: An application such as email can take several seconds to deliver a message, with delays introduced by servers, switches and routers, before it lands in the recipient's inbox.

For our specific application, latency is typically measured by sending traffic between two endpoints that are

IR sensor and information signal reaches to controller and 8 bit data send to application over serial communication.



Touch Less Elevator Panel design

Power consumption –

- Arduino UNO- 250mW
- IR Sensor-74mW, for 5 sensors-370mW
- SD card module-100mW
- Audio speaker module with amplifier- 24mW
- Total Power consumption of module-744mW

Problem Statement & Resolved State-

- Project elevator panel design and implementation, in this overall cycle the real problem statement we have found is,
  Implementation of audio voice output, as we have lack of memory in controller, so were not able to load more information which was needed in audio file format in controller.
  Solution proposed:
- We have started implementing the audio file by truncated formats which would compact in our controller, but we have had no successes in this try,
- Second, we have decided to purchase the Aurdino UNO with inbuilt SD card option, that would settle high project cost, so we left off this try,
- Finally we have decided to purchase SD card module and integrate with controller by SPI communication.
- That's made us achieving the good result in terms of required audio file format play when elevator level set.

E. Scope & Enhancement phase (Scalability)

Slno	Scope	Implemented	Future scope
1	IR sensor integration	√	
2	Audio output by Speaker	√	
3	SD card interface by SPI	√	
4	Obstacle detection and digital output	√	
5	Audio length implementation with memory card	√	
6	Priority set for the levels		x
7	Data logger		x
8	GUI	√	
9	Multiple input and stocking the value		x
10	Reset option	√	
11	Emergency call option with GPS communication		x

F. Test results

Slno	Test cases	Tested
1	Ensure physical hardware connections are appropriate with power, vcc, gnd.	Pass
2	Ensure In and Out pins are terminated correctly	Pass
3	Sensor 1 should set when an obstacle has detected and Arduino controller set by Analog input 0	Pass
4	Sensor 2 should set when an obstacle has detected and Arduino controller set by Analog input 1	Pass
5	Sensor 3 should set when an obstacle has detected and Arduino controller set by Analog input 2	Pass
6	Sensor 4 should set when an obstacle has detected and Arduino controller set by Analog input 3	Pass
7	Sensor 5 should set when an obstacle has detected and Arduino controller set by Analog input 4	Pass
8	Sensor 6 should set when an obstacle has detected and Arduino controller set by Analog input 5_Alarm state	Pass
9	Ensure SD card is accessible and store high memory data (audio file) and ensure because of large memory consumption, legacy logics are not corrupted	Pass
10	Ensure SPI communication works with SD card CS, CLK, MISO, MISI	Pass
11	Ensure sensor is active when it has detected obstacle and make sure audio output is On with appropriate audio output	Pass
12	Ensure when there is an power loss, sensor data's reset and does not occupy the value	Pass
13	Power series to be connected with all components and ensure all components are terminated and measuring <5v	Pass
14	Ensure sensor output in binary by serial port monitor and make sure each sensor has Binary output, this is what we used in GUI directions for user	Pass
15	Make sure Click here reset, button resetting the present or current level to reset and setting to initial level	Pass
16	Ensure alarm state has Sensor (red) and moveable object stops and does not hide or move to initial condition	Pass
17	Ensure after any level sensed, moveable object moves to the appropriate level and it should defined in indication by indicator box as well as Green color	Pass
18	When elevator is active (means, powered) then all sensors should wait for an input indicator should reflect by Yellow color	Pass
19	Make sure when level is crossed, yellow color should change to the sensor pitch, it ensure the level is reached and moved to next level	Pass
20	Ensure Power reset does not "Collapse" GUI and Giving multiple sensor input detection does not takes GUI to false condition	Pass

G. *References*

- IR sensor characteristics,

https://mmrc.caltech.edu/FTIR/Literature/General/Hamamatus%20IR%20dectors.pdf

- SD card characteristics,

https://www.convict.lu/pdf/ProdManualSDCardv1.9.pdf

- Arduino,

https://www.arduino.cc/en/main/boards

- 8051 controller architecture,

https://www.elprocus.com/8051-microcontroller-architecture-and-applications/