# Elevator Simulator

This is the documentation for the CSE 251 Elevator Simulator. This describes the function calls necessary to utilize the simulator. The Elevator Simulator is a graphical representation of an elevator in a three story building. Support is provided for motor control, position sensing, button support, and indicators.

See the documentation for the **[ElevatorLib.h](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html" \o "Elevator Simulator library include file.)** file for all of the function documentation. Applications include this file to gain access to the functions and link to the file libElevatorLib.a. A makefile is provided with the project.

This documentation is generated from sources using Doxygen.

**ElevatorLib.h File Reference**

Elevator Simulator library include file. [More...](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#_details)

#include <stdbool.h>  
#include <unistd.h>

[Go to the source code of this file.](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h_source.html)

|  |  |
| --- | --- |
| **Defines** | |
| #define | [**FloorSpacing**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a57f1811926b159451184a4631fb767ed)   3.28 |
|  | Distance from floor to floor in meters. |
| #define | [**FloorTolerance**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#af359cf93a22e275c189ece93681fac4e)   0.16 |
|  | Tolerance for a safe door opening. |
| **Functions** | |
| bool | [**IsElevatorRunning**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a20795ced0bb54785da982c2b418a8d17) (void) |
|  | Function to determine if the elevator simulator is currently running. |
| void | [**ElevatorStartup**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#aeea8ff2ded071db73fccf672417b8cc2) (void) |
|  | Start the Elevator Simulation. |
| void | [**ElevatorShutdown**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a66a8a56904d41ab2d0ec07ce459505ce) (void) |
|  | Shut down the Elevator Simulation. |
| void | [**SetCloseDoorLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a5400d6d654a4823920359c27b5c1f4f2) (bool s) |
|  | Set the status of the close door light on the panel in the elevator car. |
| bool | [**GetCloseDoorLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#ab25a39d79643647e038ed0fab0a92926) (void) |
|  | Get the status of the close door light on the panel in the elevator car. |
| void | [**SetOpenDoorLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a57b73797cfbe4c847a5f50bf17da856b) (bool s) |
|  | Set the status of the open door light on the panel in the elevator car. |
| bool | [**GetOpenDoorLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a2aaa63011ec20b5bff6fce543973ca83) (void) |
|  | Get the status of the open door light on the panel in the elevator car. |
| void | [**SetPanelFloorLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a1d3424c3328dd3a1684a8c099ad4df80) (int floor, bool s) |
|  | Set the status of the floor select light on the panel in the elevator car. |
| bool | [**GetPanelFloorLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#ac69a408c736689cf5ed3b53f508334a2) (int floor) |
|  | Get the status of the floor select light on the panel in the elevator car. |
| void | [**SetCallLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a100483e56e0d71ab88fc7b3d2483506b) (int floor, bool up, bool s) |
|  | Set the status of a call button light for a floor. |
| bool | [**GetCallLight**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a16b6252a2f48e592ea535f956a839231) (int floor, bool up) |
|  | Get the status of a call button light for a floor. |
| void | [**SetDoorIndicator**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a6f84528ef3cb7cb400deec2d88393942) (int floor, bool up, bool s) |
|  | Set the status of a door indicator for a floor. |
| bool | [**GetDoorIndicator**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a981fc47866a4ad067b213371cbffa380) (int floor, bool up) |
|  | Get the status of a door indicator light for a floor. |
| void | [**SetDoor**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#acf6ea341b912ff34b94ff6910bfcf825) (int floor, bool open) |
|  | Set the opening/closing status of the elevator doors. |
| bool | [**IsDoorOpen**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#aaf6a02fd80343cf80b20530c8942c4f2) (int floor) |
|  | Determines if the elevator door is open. |
| bool | [**IsDoorClosed**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a51adeab0ffb55cb7af51f62c1df5d485) (int floor) |
|  | Determines if the elevator door is closed. |
| void | [**SetOpenDoorHandler**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a85d94dc86ffd940551a8074f58e3e35f) (void(\*handler)()) |
|  | Set a function to be called when the door open button is pressed. |
| void | [**SetCloseDoorHandler**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a7b74b42732e5a5ae4cdc29924f356e2f) (void(\*handler)()) |
|  | Set a function to be called when the door close button is pressed. |
| void | [**SetCallPressedHandler**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a78a6da25c4ba254f231129038d25e670) (void(\*handler)(int floor, bool up)) |
|  | Set a function to be called when a call button on a floor is pressed. |
| void | [**SetFloorPressedHandler**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a8652216b1b47cf71bc254cce43d64516) (void(\*handler)(int floor)) |
|  | Set a function to be called when a floor button on the elevator car panel is pressed. |
| void | [**SetBrake**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a951bc61e36d6e40f01972e5b7e04d242) (bool b) |
|  | Set the status of the elevator brake. |
| void | [**SetMotorPower**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#af81062347eed14f0f5cc4169bff9f56c) (double power) |
|  | Set the motor power. |
| double | [**GetPosition**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a00ddb83939d5c30404b30667dca980b2) (void) |
|  | Get the absolute position of the elevator in meters. |
| double | [**GetVelocity**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#ab29287a794268e2008171fbd626cef2d) (void) |
|  | Get the velocity of the elevator in meters per second. |
| double | [**GetFloor**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a9440c5eca5347bb7d1abf78e91f0305d) (void) |
|  | Get the current floor the elevator is on as a floating point value. |
| int | [**GetNearestFloor**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#ae3c277b47513204974117b39002c5b3d) (void) |
|  | Get the nearest integer floor to the elevator. |
| void | [**ChangeLoading**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a37a903320308b9dbbab6945eb04d7004) (void) |
|  | Change the current number of passengers on the elevator. |
| void | [**SetLoading**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#abfbd19923fed0da6f40d56fe791c4b6f) (int r) |
|  | Set the passenger load for the elevator car. |
| void | [**ResetTimer**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a2edddcb800956f75e21762453af76ed8) (void) |
|  | Reset a timer available for program use. |
| double | [**GetTimer**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a18d6f64da6d0af0ec85a1f1e0264f8b4) (void) |
|  | Read the current value of the timer. |
| int | [**WhatFloorToGoTo**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#acbbd4ae93ea54fc6e4428b19bccf2e9d) (bool up) |
|  | Determine what floor the elevator should be going to. |
| int | [**WhatFloorToGoToInThisDirection**](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a9065c2c9c5b9446022b258ecc61a4ffd) (bool up) |
|  | Determine what floor the elevator should be going to when traveling in a certain direction. |

**Detailed Description**

Elevator Simulator library include file.

This file is included to utilize the Elevator Simulator.

**Define Documentation**

|  |
| --- |
| **#define FloorTolerance   0.16** |

Tolerance for a safe door opening.

If the car is within 16cm of a floor, the door opening is considered safe. Farther away and we have a safety issue.

**Function Documentation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void ChangeLoading** | **(** | **void** |  | **)** |  |

Change the current number of passengers on the elevator.

This function will cause the number of passengers on the elevator to change to a new random number. This happens automatically when the door opens on any floor, but can also be used to test elevator control code by selecting random loadings.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void ElevatorShutdown** | **(** | **void** |  | **)** |  |

Shut down the Elevator Simulation.

This function should be called last in the program. If the simulator is currently running, it shuts down the simulator and closes the window. Then it releases any resources allocated to this program.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void ElevatorStartup** | **(** | **void** |  | **)** |  |

Start the Elevator Simulation.

This function must be called before any other call can be made to any Elevator Simulator functions. It starts the elevator simulator in a new window.

|  |  |  |  |
| --- | --- | --- | --- |
| **bool GetCallLight** | **(** | **int** | **floor,** |
|  |  | **bool** | **up** |  |
|  | **)** |  |  |  |

Get the status of a call button light for a floor.

This functions checks to see if an elevator call button for a floor is currently lit. Each floor has an up and down call button. This function determines the status of the lighting of those buttons.

The default button behavior sets this light on when the button is pressed (except for down on the bottom floor and top on the upper floor, where the button press is ignored). Use SetCallLight to turn the light back off.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. |
|  | *up* | If true, the up button is read. If false, the down button is read. |
|  | *s* | Set to true to set the light. |

**Returns:**

true if the light is on.

if(GetCallLight(1, false))

{

do something here...

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **bool GetCloseDoorLight** | **(** | **void** |  | **)** |  |

Get the status of the close door light on the panel in the elevator car.

This functions checks to see if the close door light on the panel in the elevator car is on.

**Returns:**

true if the light is on.

if(GetCloseDoorLight())

{

do something here...

}

|  |  |  |  |
| --- | --- | --- | --- |
| **bool GetDoorIndicator** | **(** | **int** | **floor,** |
|  |  | **bool** | **up** |  |
|  | **)** |  |  |  |

Get the status of a door indicator light for a floor.

This functions checks to see if a door indicator for a floor is currently lit. Each floor has an up and down arrow above the elevator door. This light is normally used to indicate when the elevator has arrived and which direction it is going. This function checks to see if those indicators are on or off.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. |
|  | *up* | If true, the up indicator is read. If false, the down indicator is read. |

**Returns:**

true if the light is on.

if(GetDoorIndicator((3, false))

{

do something here...

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **double GetFloor** | **(** | **void** |  | **)** |  |

Get the current floor the elevator is on as a floating point value.

This function returns the position of the elevator car in floors. A return value of 1 indicates the elevator is at the first floor door. A value of 3 indicates the elevator is on the third floor door. This is a floating point value and will indicate locations between floors.

This function return value is equivalent to: **[GetPosition()](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html" \l "a00ddb83939d5c30404b30667dca980b2" \o "Get the absolute position of the elevator in meters.)** / FloorSpacing + 1

**Returns:**

The current floor as a floating point value.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **int GetNearestFloor** | **(** | **void** |  | **)** |  |

Get the nearest integer floor to the elevator.

This function returns floor the elevator car is nearest as an integer value. A return value of 1 indicates the elevator is nearest to the first floor. A value of 3 indicates the elevator is nearest to the third floor.

This function return value is equivalent to the value of **[GetFloor()](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html" \l "a9440c5eca5347bb7d1abf78e91f0305d" \o "Get the current floor the elevator is on as a floating point value.)** rounded to the nearest integer.

**Returns:**

The nearest floor as an integer value.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **bool GetOpenDoorLight** | **(** | **void** |  | **)** |  |

Get the status of the open door light on the panel in the elevator car.

This functions checks to see if the open door light on the panel in the elevator car is on.

**Returns:**

true if the light is on.

if(GetOpenDoorLight())

{

do something here...

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **bool GetPanelFloorLight** | **(** | **int** | **floor** | **)** |  |

Get the status of the floor select light on the panel in the elevator car.

This functions checks to see if the floor select button on the panel in the elevator car is currently lit.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. |

**Returns:**

true if the light is on.

if(GetPanelFloorLight(1))

{

do something here...

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **double GetPosition** | **(** | **void** |  | **)** |  |

Get the absolute position of the elevator in meters.

This function returns the position of the elevator in the shaft in meters relative to the first floor. A value of zero means the elevator is at the first floor door. The constant FloorSpacing indicates the spacing between floors. If this function returns the value in FloorSpacing, the elevator is at the second floor. This is a continuous value and can be negative if the elevator is below the first floor door.

**Returns:**

The current position in meters relative to the first floor door.

if(GetPosition() > [FloorSpacing](http://www.cse.msu.edu/~cse251/project2/doc/html/_elevator_lib_8h.html#a57f1811926b159451184a4631fb767ed))

{

printf("The elevator is above the second floor\n");

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **double GetTimer** | **(** | **void** |  | **)** |  |

Read the current value of the timer.

This function read the current value of an available timer. The timer can be reset using the ResetTimer call. The value read is in second and is a floating point value.

**Returns:**

Number of seconds since the last time ResetTimer was called.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **double GetVelocity** | **(** | **void** |  | **)** |  |

Get the velocity of the elevator in meters per second.

This function returns the current velocity of the elevator car in meters per second. A positive value indicates the elevator is going up. A negative value indicates the elevator is going down. Do not exceed the maximum design velocity of 1 meter per second in either direction.

**Returns:**

The elevator car velocity in meters per second

if(fabs(GetVelocity()) < 0.33)

{

printf("It is okay to brake\n");

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **bool IsDoorClosed** | **(** | **int** | **floor** | **)** |  |

Determines if the elevator door is closed.

This function returns true if the elevator door is completely closed. When a door is moving from open to closed or vice versa, this function and IsDoorOpen will both return false. Use this function to determine when the closing process has completed.

**Parameters:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. | | | | | | |
| **bool IsDoorOpen** | | | | **(** | **int** | **floor** | **)** |  |

Determines if the elevator door is open.

This function returns true if the elevator door is completely open. When a door is moving from open to closed or vice versa, this function and IsDoorClosed will both return false. Use this function to determine when the opening process has completed.

**Parameters:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. | | | | | | |
| **bool IsElevatorRunning** | | | | **(** | **void** |  | **)** |  |

Function to determine if the elevator simulator is currently running.

This function should be called in a loop that exists when the elevator is no longer running. If the elevator window is closed by the user, the calls are not longer valid and should not be used (other than ElevatorShutdown).

**Returns:**

true if the elevator simulation is currently running.

while(IsElevatorRunning())

{

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void ResetTimer** | **(** | **void** |  | **)** |  |

Reset a timer available for program use.

This function, when called, resets a timer to zero. It is equivalent to pressing the reset button on a stopwatch. The stopwatch can then be read using the GetTimer function. There is only one timer available to an application.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void SetBrake** | **(** | **bool** | **b** | **)** |  |

Set the status of the elevator brake.

This function controls the elevator brake. It called with true, the brake is set on. If called with false, the brake is set off. The brake will immediately stop the movement of the car unless the car is going faster than 0.33 meters per second. If the car is going faster than 0.33 meters per second, indicated by the velocity outside the green on the speed indicator, the brake will be destroyed and will no longer work.

**Parameters:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *b* | The brake status to be set. true enables the brake.  SetBrake(true); | | | | | |
| **void SetCallLight** | | | | **(** | **int** | **floor,** |
|  | | | |  | **bool** | **up,** |
|  | | | |  | **bool** | **s** |  |
|  | | | | **)** |  |  |  |

Set the status of a call button light for a floor.

This function sets the lighted status of a call button on a floor. Each floor has an up and down call button. This sets the lighting of that button on or off.

**Parameters:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. | | | |
|  | *up* | If true, the up button is set. If false, the down button is set. | | | |
|  | *s* | Set to true to turn the light on  SetCallLight(2, true, true); | | | |
| **void SetCallPressedHandler** | | | | **(** | **void(\*)(int floor, bool up)** | **handler** | **)** |  |

Set a function to be called when a call button on a floor is pressed.

This function installs a handler function that will be called when a elevator call button on any floor is pressed.

If no handler is installed, the default behavior for a call button press is to light the call button on that floor, with the exception that the up button on the top floor and the down button on the bottom floor do nothing. If a handler is installed, this default behavior does not occur.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *handler* | A C function that will be called when the button is pressed. |

**Example of how to set the handler in code:**

SetCallPressedHandler(OnCallPressed);

**Example handler function:**

void OnCallPressed(int floor, bool up)

{

if(up)

{

printf("The up call button has been pressed on floor %d\n", floor);

}

else

{

printf("The down call button has been pressed on floor %d\n", floor);

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void SetCloseDoorHandler** | **(** | **void(\*)()** | **handler** | **)** |  |

Set a function to be called when the door close button is pressed.

This function installs a handler function that will be called when the close door button on the panel inside the elevator is pressed.

There is no default behavior for this button.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *handler* | A C function that will be called when the button is pressed. |

**Example of how to set the handler in code:**

SetCloseDoorHandler(OnOpenDoor);

**Example handler function:**

void OnCloseDoor()

{

printf("Close door pressed\n");

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void SetCloseDoorLight** | **(** | **bool** | **s** | **)** |  |

Set the status of the close door light on the panel in the elevator car.

This function sets the lighted status of the close door button inside the elevator car. If set to true, the light is turned on.

**Parameters:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *s* | Set to true to run the light on  SetCloseDoorLight(true); | | | | | |
| **void SetDoor** | | | | **(** | **int** | **floor,** |
|  | | | |  | **bool** | **open** |  |
|  | | | | **)** |  |  |  |

Set the opening/closing status of the elevator doors.

The elevator doors have automatic control hardware that opens and closes then based on this call. If you call this function with open set true, the door will begin to open. That opening takes a finite amount of time, so it is not immediately open. If the door is closing and this call requests an open, it will reverse and begin opening again. The same is true if opening.

**It is the control system's responsibility to only open doors**

when the elevator is at the floor and not to move the elevator while the doors are open.

**Parameters:**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. | | | | | |
|  | *open* | true tells the doors to open on that floor. | | | | | |
| **void SetDoorIndicator** | | | | **(** | **int** | **floor,** |
|  | | | |  | **bool** | **up,** |
|  | | | |  | **bool** | **s** |  |
|  | | | | **)** |  |  |  |

Set the status of a door indicator for a floor.

This function sets the lighted status of a door indicator on a floor. Each floor has an up and down arrow above the elevator door. This light is normally used to indicate when the elevator has arrived and which direction it is going. This function turns those indicators on and off.

**Parameters:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. | | | | | | |
|  | *up* | If true, the up arrow is set. If false, the down arrow is set. | | | | | | |
|  | *s* | Set to true to turn the light on  SetCallLight(2, true, true); | | | | | | |
| **void SetFloorPressedHandler** | | | | **(** | **void(\*)(int floor)** | **handler** | **)** |  |

Set a function to be called when a floor button on the elevator car panel is pressed.

This function installs a handler function that will be called when a floor select button is pressed on the elevator panel.

If no handler is installed, the default behavior for a floor button press is to light the button for that floor on the elevator panel. If a handler is installed, this default behavior does not occur.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *handler* | A C function that will be called when the button is pressed. |

**Example of how to set the handler in code:**

SetFloorPressedHandler(OnFloorPressed);

**Example handler function:**

void OnFloorPressed(int floor)

{

printf("The elevator panel button has been pressed for floor %d\n", floor);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void SetLoading** | **(** | **int** | **r** | **)** |  |

Set the passenger load for the elevator car.

This function sets the passengers in the elevator car. The passengers are represented by a bitwise or of 1 for Noel, 2 for Neil, and 4 for Sparty. To turn on all passengers, use a value of 1 | 2 | 4 = 7.

**Parameters:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *r* | Binary or of passenger values. This code example enables all passengers.  SetLoading(7); | | | | | | |
| **void SetMotorPower** | | | | **(** | **double** | **power** | **)** |  |

Set the motor power.

The elevator uses a DC motor. This function sets how much power is provided to the motor. The values range from -1 to 1. A value of -1 means maximum velocity down. A value of 1 means maximum velocity up. Values outside this range will be bounded to the range [-1, 1]. Motor power translates to torque and is not an indicator of speed.

**Parameters:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *power* | The power setting in the range -1 to 1.  SetMotorPower(0.4); | | | | |
| **void SetOpenDoorHandler** | | | | **(** | **void(\*)()** | **handler** | **)** | |  |

Set a function to be called when the door open button is pressed.

This function installs a handler function that will be called when the open door button on the panel inside the elevator is pressed.

There is no default behavior for this button.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *handler* | A C function that will be called when the button is pressed. |

**Example of how to set the handler in code:**

SetOpenDoorHandler(OnOpenDoor);

**Example handler function:**

void OnOpenDoor()

{

printf("Open door pressed\n");

}

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **void SetOpenDoorLight** | **(** | **bool** | **s** | **)** |  |

Set the status of the open door light on the panel in the elevator car.

This function sets the lighted status of the open door button inside the elevator car. If set to true, the light is turned on.

**Parameters:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | *s* | Set to true to run the light on  SetOpenDoorLight(true); | | | | |
| **void SetPanelFloorLight** | | | | **(** | **int** | **floor,** |
|  | | | |  | **bool** | **s** |  | |
|  | | | | **)** |  |  |  | |

Set the status of the floor select light on the panel in the elevator car.

This function sets the lighted status of the floor select button inside the elevator car. If set to true, the light is turned on.

**Parameters:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *floor* | The floor number. 1, 2, or 3. | | | | | | |
|  | *s* | Set to true to run the light on  SetPanelFloorLight(1, true); | | | | | | |
| **int WhatFloorToGoTo** | | | | **(** | **bool** | **up** | **)** |  |

Determine what floor the elevator should be going to.

This function will check the elevator buttons and determine what floor the elevator should be going to. When going up, it will look for floors above the current floor that have an up call button or elevator panel button pressed. If so, it will return the nearest one. Otherwise, it will check for floors above the current floor with a down button pressed. If it finds one, it will return the farthest of these. If it finds no floors in the up direction, it repeats the process in the down direction.

The process when going down is similar, mirroring the up and down states.

This function is usually called when the elevator is stopped to determine where it should go next.

This function is implemented by calling WhatFloorToGoToInThisDirection first with the value of up, then, if no floor is selected, with the value of !up.

**Parameters:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *up* | If true, call buttons and the elevator buttons are first checked in the up direction, then down. If no floors are selected, the function returns a value of -1. Return values for floors are 1, 2, or 3. | | | | | | |
| **int WhatFloorToGoToInThisDirection** | | | | **(** | **bool** | **up** | **)** |  |

Determine what floor the elevator should be going to when traveling in a certain direction.

This function will check the elevator buttons and determine what floor the elevator should be going to. When going up, it will look for floors above the current floor that have an up call button or elevator panel button pressed. If so, it will return the nearest one. Otherwise, it will check for floors above the current floor with a down button pressed. If it finds one, it will return the farthest of these. If it finds no floors in the up direction, it returns a value of 3, so you will always have a floor to go to.

The process when going down is similar, mirroring the up and down states.

This function can be continuously called while the elevator is moving to determine what floor it should stop on. The function will not return a floor that the elevator does not have time to safely stop for. For example, if the elevator is going up and is just below the second floor when the call button for the second floor is pressed, the function will not return the second floor because the elevator could not stop quickly enough to arrive at that floor.

**Parameters:**

|  |  |  |
| --- | --- | --- |
|  | *up* | If true, call buttons and the elevator buttons are checked in the up direction. If no floors are selected, the function returns a value of 3 if up is true and 1 if up is false. Return values for floors are 1, 2, or 3. |