Co-funded by the
Erasmus+ Programme
of the European Union

SEAS 4.0

SUSTAINABLE
SHIP
AND SHIPPING
4.0
Erasmus Mundus Joint Master
Degree

Excellence in education for a new generation of Naval Architects and Marine Engineers

INDUSTRY 4.0 ENABLING TECHNOLOGIES

Smart Contract for Incentivizing Valid Usage of Energy

Team
Abisade Folarin
Ahsan Ullah Jawad
Muhammad Fuad Shofly
Pritam Kumar Patel
Saeid Panahi

Course Teacher
Professor Dr. Tiago M. Fernández
Professor Dr. Paula Fraga Lamas

1. Introduction

As a continuation of the IoT assignment on Intelligent lighting system of the Industry 4.0 Enabling Technologies course, a smart contract has been written based on Ethereum to send incentives for valid usage of energy. In this contract, all sensor input can be stored with a unique id, various valid conditions can be checked, and Wei can be sent inward and outward the contract if a certain condition is met.

2. Case

Numerous light sources are the substantial part of the energy consumption of a ship. It has been observed that reduction of the usage of these lights can significantly reduce energy consumption, carbon emissions, and light pollution, as well as increase the longevity of the devices leading to achieve a sustainable society. Apart from the personal or organizational urge, monetary incentive has always been proven as an effective tool to achieve a target. Hence, a smart contract has been written to send incentives only for valid usage of light sources.

3. Users

Resembling a regular contract, this smart contract has two parties. One is the ship crew, and the other is the ship owner.

4. Technical Parameter

This smart contract has been written in the Solidity programming language that runs on Ethereum Virtual Machine. To compile the code Remix IDE has been used and the code has been run on JavaScript VM(London) environment.

5. Contents

The contract is focused on detecting the necessary usage of the light source. Only in case of a valid reason for light source usage, the ship owner can send incentives for example WEI to the ship crew.

In the ship, Push button, RTC, LDR, and PIR sensors will gather the information about the surrounding. The algorithm of the code will confirm the validity. Valid reasons include emergency, nighttime, darkness, and the presence of a human. All these sensors generate the value along with the status of the light source. These values are the input of the code. A struct "Incident" will gather all the sensors' value. Each time these inputs are logged, "generateincidentId" will create a unique id using the hash function. The function "new_incident" will create an incident and add each of the incidents into the array of the "Incident" struct. An Event "incident_register" will emit each time a incident is logged along with the index number. The shipowner can see the input log history using the index number.

A function “check_validity” contains the if-else condition that checks the validity of the led usage. There are a few cases when all sensors’ data might not be necessary. In such a case, the value of the sensors should be “-1”. The function “Validity” will show the validity of the input.

A smart contract is more impactful when it can transact money i.e., Ether in this case. Therefore, the ship owner can send Wei from his account to this contract using the “receivewei” function. Each time of transection the amount of the Wei should be a minimum of 500. If the sender sends less than this value transaction will be failed displaying an error message. Each time the owner sends money to the contract, it will be added to the current balance.

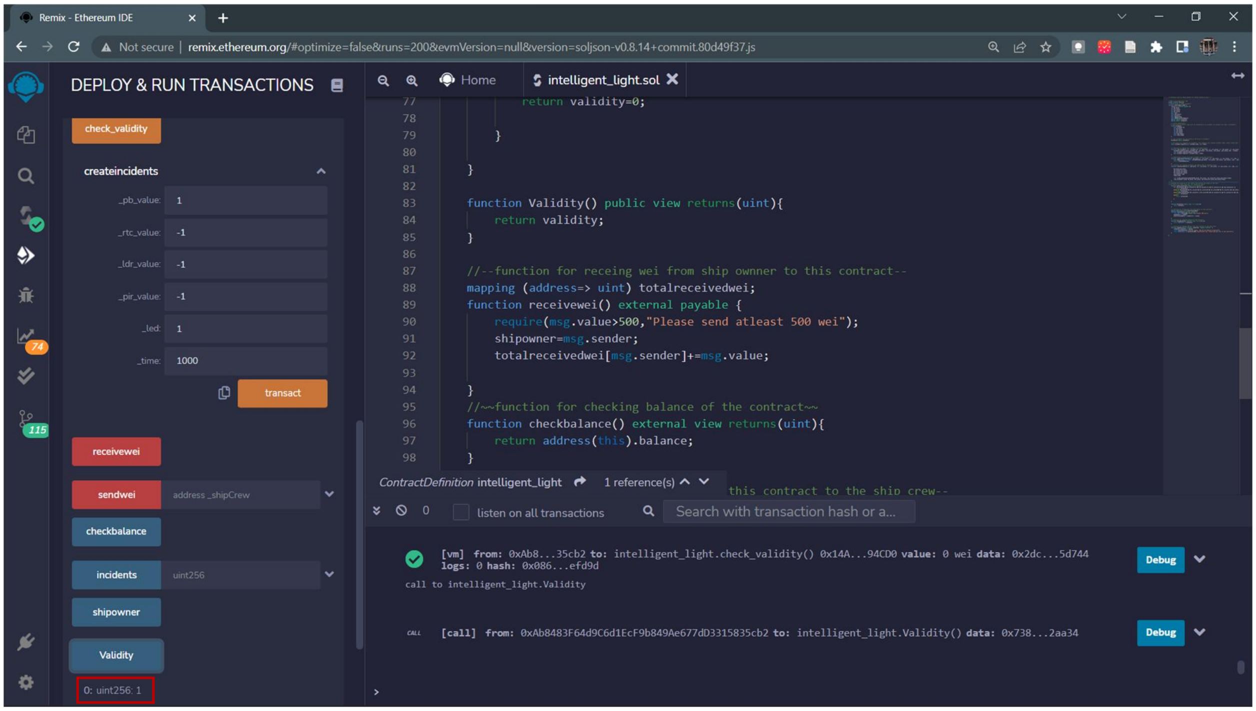
To check the current balance of the contract, the “checkbalance” function can be used.

Finally, the ship owner can send the Wei to the ship crew using “sendwei” function. This function only executes the transaction if there is a valid reason for energy usage. An error will be displayed with a message if the reason is invalid. Moreover, a minimum balance should be more than 100 Wei, otherwise, the transaction will not be executed along with another error message.

6. Testing

The code has been tested on different scenarios as mentioned below.

a) **<u>Scenario 1:</u>** Push Button: On, Disregard other sensors, Led: On, Reason: Valid


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
check_validity
createincidents
transact
receivewei
sendwei
checkbalance
incidents
shipowner
Validity
Home
intelligent_light.sol
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
Debug

_pb_value: 1
_rtc_value: -1
_ldr_value: -1
_pir_value: -1
_led: 1
_time: 1000

sendwei: address _shipCrew
incidents: uint256

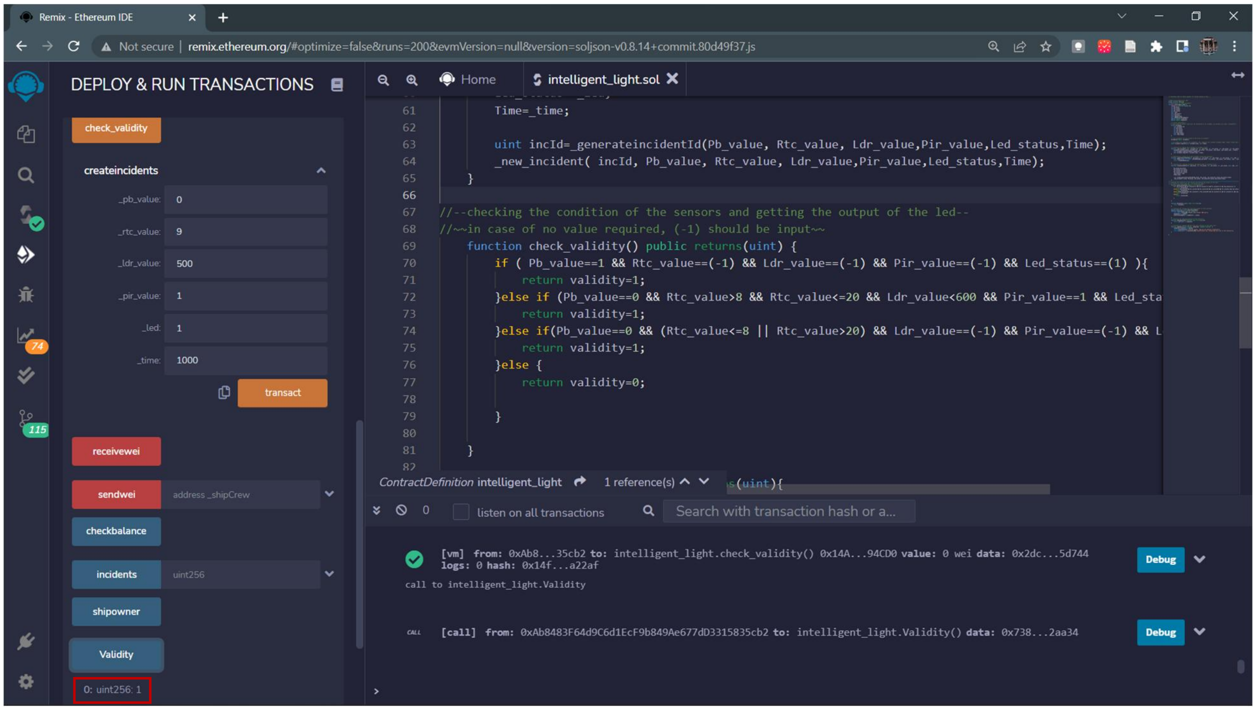
0: uint256: 1

```
            return validity=0;

        }

    }

    function Validity() public view returns(uint){
        return validity;
    }

    //--function for receing wei from ship ownner to this contract--
    mapping (address=> uint) totalreceivedwei;
    function receivewei() external payable {
        require(msg.value>500,"Please send atleast 500 wei");
        shipowner=msg.sender;
        totalreceivedwei[msg.sender]+=msg.value;

    }
    //~~function for checking balance of the contract~~
    function checkbalance() external view returns(uint){
        return address(this).balance;
    }
```

```
this contract to the ship crew--
```

```
[vm] from: 0xAb8...35cb2 to: intelligent_light.check_validity() 0x14A...94CD0 value: 0 wei data: 0x2dc...5d744
logs: 0 hash: 0x086...efd9d
call to intelligent_light.Validity

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.Validity() data: 0x738...2aa34
```

b) **Scenario 2:** Push Button: Off, Time: Day, Light: Dark, Pir: On, Led: On, Reason: Valid


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
check_validity
createincidents
transact
receivewei
sendwei
address _shipCrew
checkbalance
incidents
uint256
shipowner
Validity
74
115
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
Debug

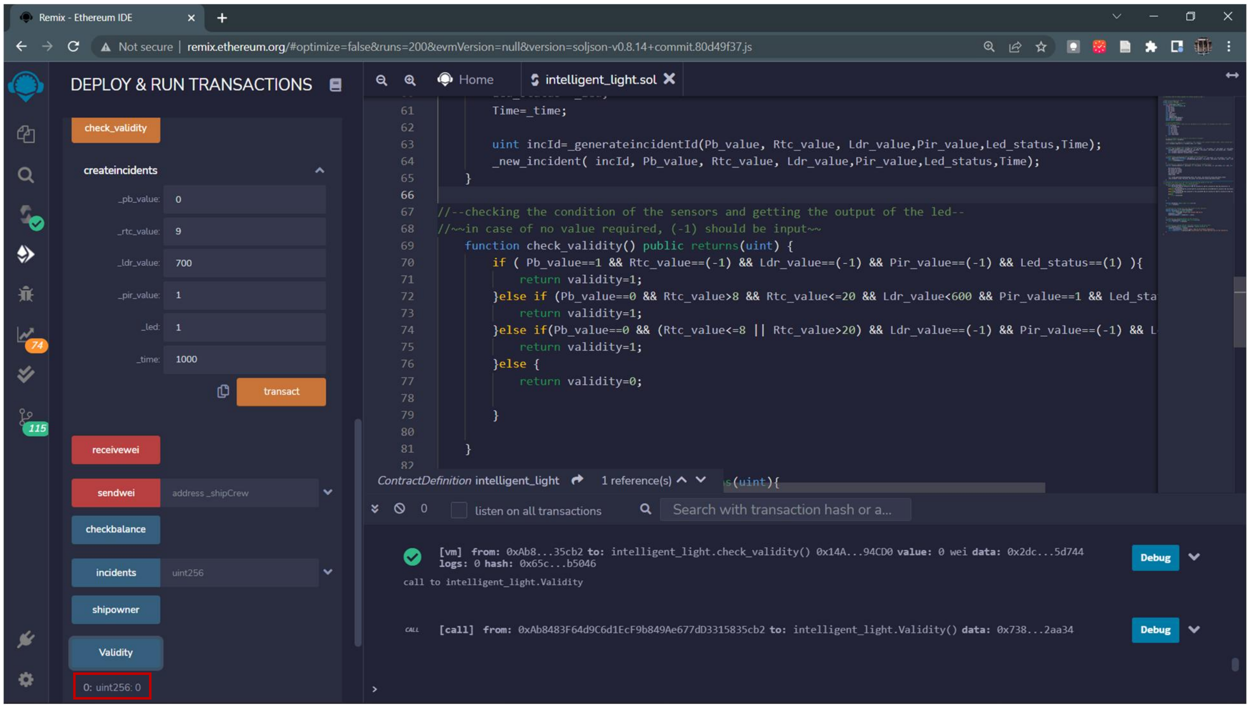
_pb_value: 0
_rtc_value: 9
_ldr_value: 500
_pir_value: 1
_led: 1
_time: 1000

0: uint256: 1

```
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

        }

    }
```

```
[vm] from: 0xAb8...35cb2 to: intelligent_light.check_validity() 0x14A...94CD0 value: 0 wei data: 0x2dc...5d744
logs: 0 hash: 0x14f...a22af
call to intelligent_light.Validity

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.Validity() data: 0x738...2aa34
```

c) **Scenario 3:** Push Button: Off, Time: Day, Light: Bright, Pir: On, Led: On, Reason: Invalid


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
check_validity
createincidents
transact
receivewei
sendwei
address _shipCrew
checkbalance
incidents
uint256
shipowner
Validity
74
115
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
Debug

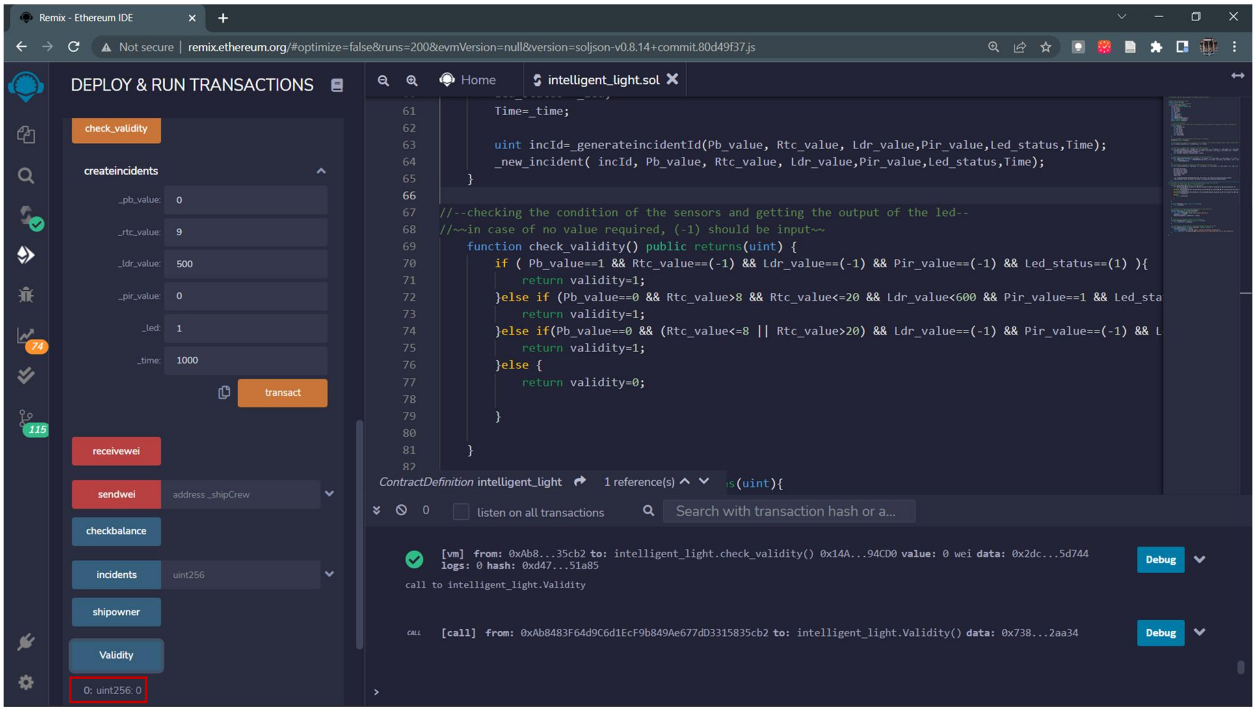
_pb_value: 0
_rtc_value: 9
_ldr_value: 700
_pir_value: 1
_led: 1
_time: 1000

0: uint256: 0

```
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

        }

    }
```

```
[vm] from: 0xAb8...35cb2 to: intelligent_light.check_validity() 0x14A...94CD0 value: 0 wei data: 0x2dc...5d744
logs: 0 hash: 0x65c...b5046
call to intelligent_light.Validity

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.Validity() data: 0x738...2aa34
```

d) **Scenario 4:** Push Button: Off, Time: Day, Light: Dark, Pir: off, Led: On, Reason: Invalid


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
check_validity
createincidents
transact
receivewei
sendwei
address _shipCrew
checkbalance
incidents
uint256
shipowner
Validity
74
115
Home
intelligent_light.sol
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
Debug

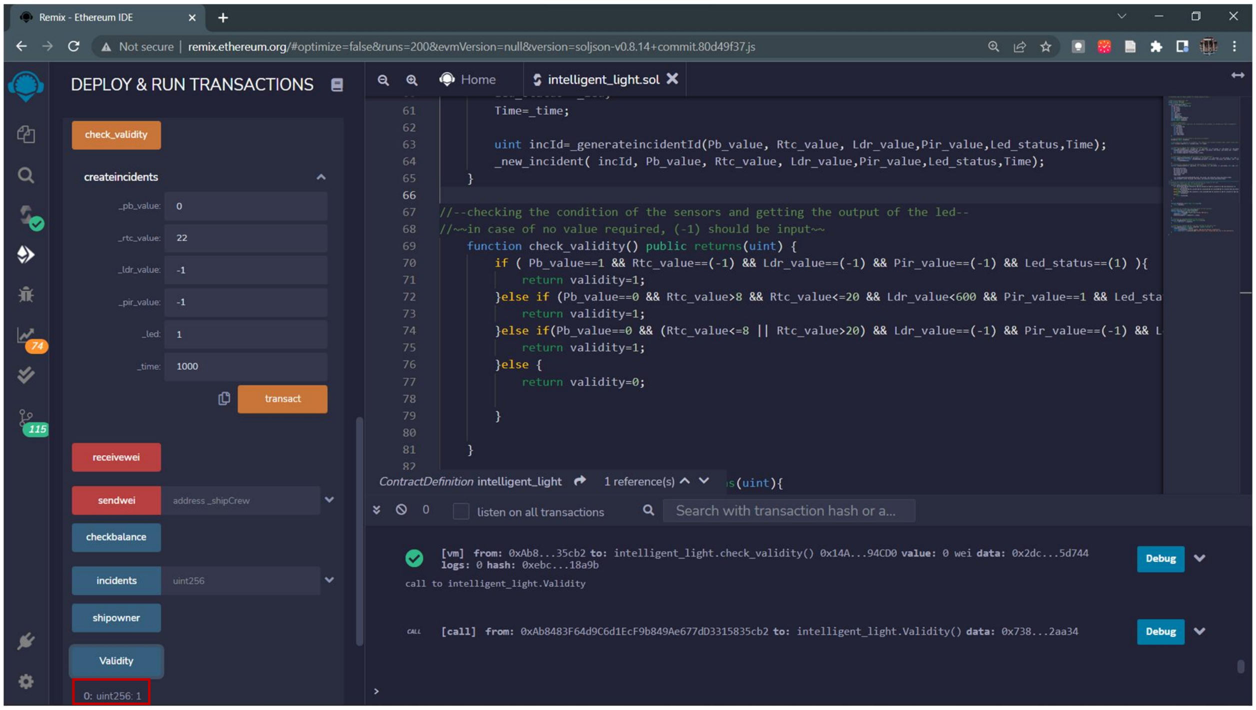
_pb_value: 0
_rtc_value: 9
_ldr_value: 500
_pir_value: 0
_led: 1
_time: 1000

0: uint256: 0

```
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

        }

    }
s(uint){
```

```
[vm] from: 0xAb8...35cb2 to: intelligent_light.check_validity() 0x14A...94CD0 value: 0 wei data: 0x2dc...5d744
logs: 0 hash: 0xd47...51a85
call to intelligent_light.Validity

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.Validity() data: 0x738...2aa34
```

e) **Scenario 5:** Push Button: Off, Time: Night, Disregard other sensors, Led: On, Reason: Valid


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
check_validity
createincidents
transact
receivewei
sendwei
address _shipCrew
checkbalance
incidents
uint256
shipowner
Validity
74
115
Home
intelligent_light.sol
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
Debug

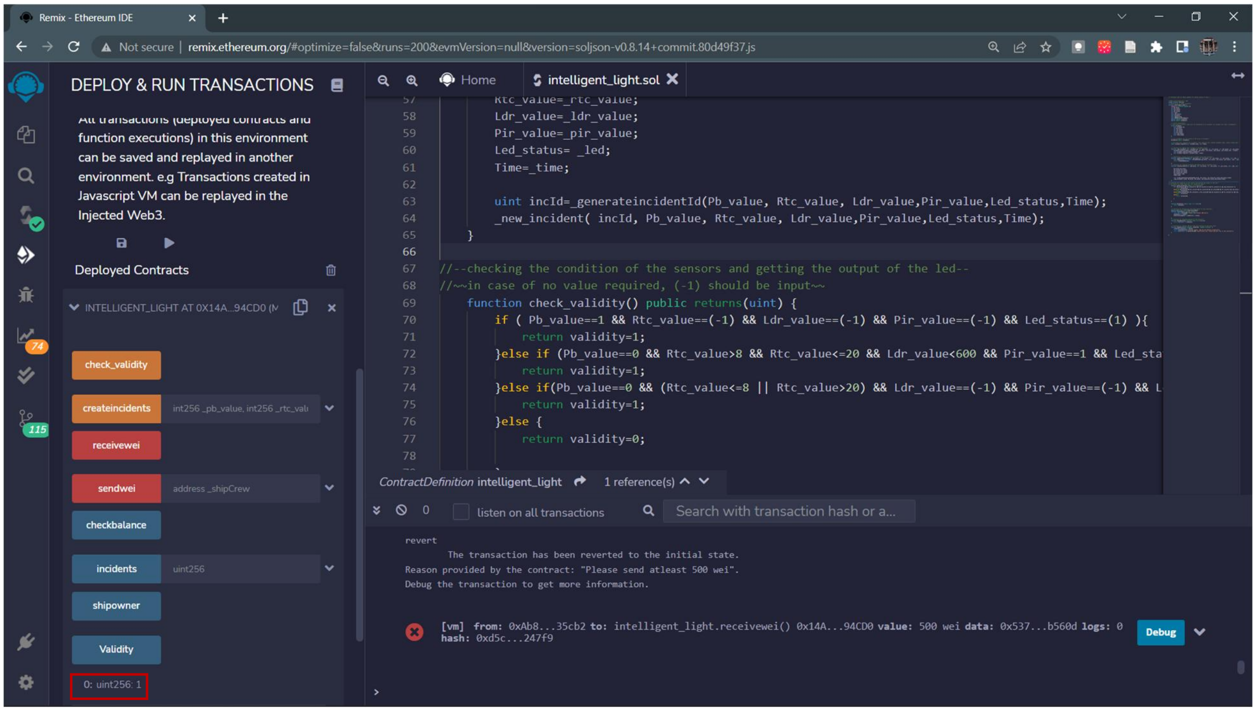
_pb_value: 0
_rtc_value: 22
_ldr_value: -1
_pir_value: -1
_led: 1
_time: 1000

0: uint256: 1

```
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

        }

    }
s(uint){
```

```
[vm] from: 0xAb8...35cb2 to: intelligent_light.check_validity() 0x14A...94CD0 value: 0 wei data: 0x2dc...5d744
logs: 0 hash: 0xebc...18a9b
call to intelligent_light.Validity

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.Validity() data: 0x738...2aa34
```

f) **Scenario 6:** Condition: Not meet, Receiving Wei: Failed


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
Deployed Contracts
INTELLIGENT_LIGHT AT 0X14A...94CD0 (M
check_validity
createincidents
int256 _pb_value, int256 _rtc_valu
receivewei
sendwei
address _shipCrew
checkbalance
incidents
uint256
shipowner
Validity
74
115
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug

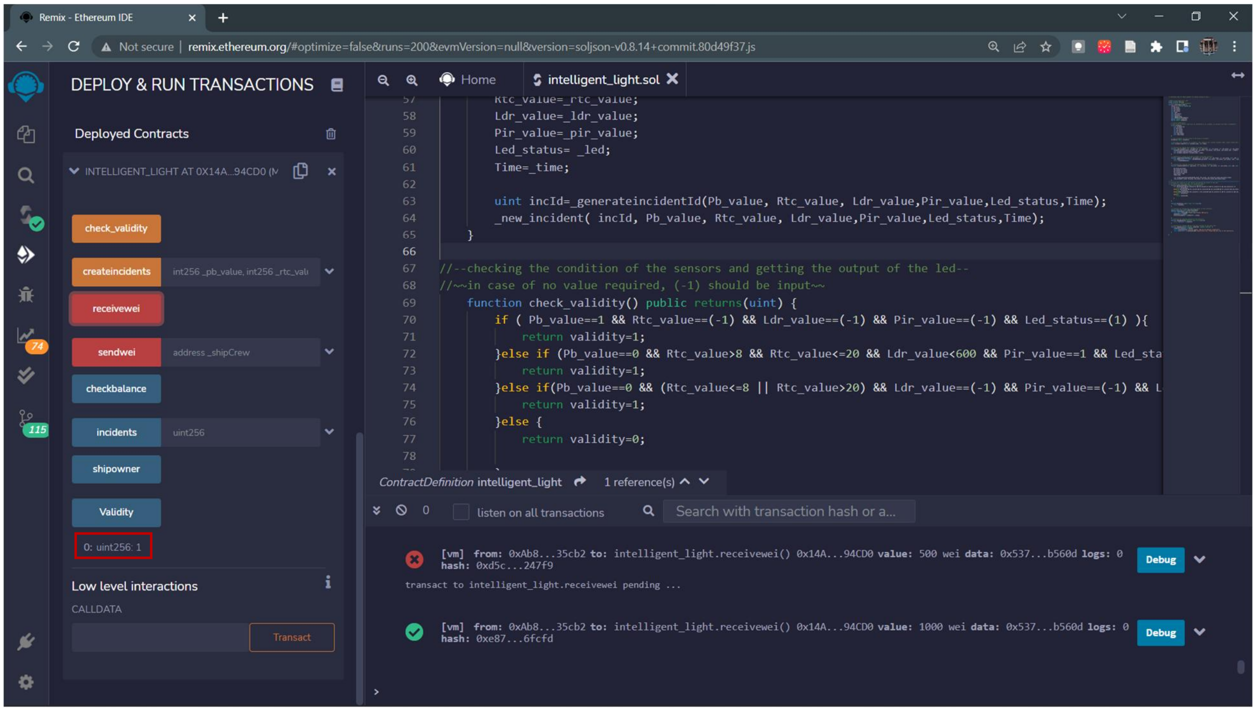
All transactions (deployed contracts and function executions) in this environment can be saved and replayed in another environment. e.g Transactions created in Javascript VM can be replayed in the Injected Web3.

0: uint256: 1

```
        Rtc_value=_rtc_value;
        Ldr_value=_ldr_value;
        Pir_value=_pir_value;
        Led_status= _led;
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

```

```
revert
        The transaction has been reverted to the initial state.
Reason provided by the contract: "Please send atleast 500 wei".
Debug the transaction to get more information.

[vm]  from: 0xAb8...35cb2 to: intelligent_light.receivewei() 0x14A...94CD0 value: 500 wei data: 0x537...b560d logs: 0
hash: 0xd5c...247f9
```

g) **Scenario 7:** Condition: meet, Receiving Wei: Executed

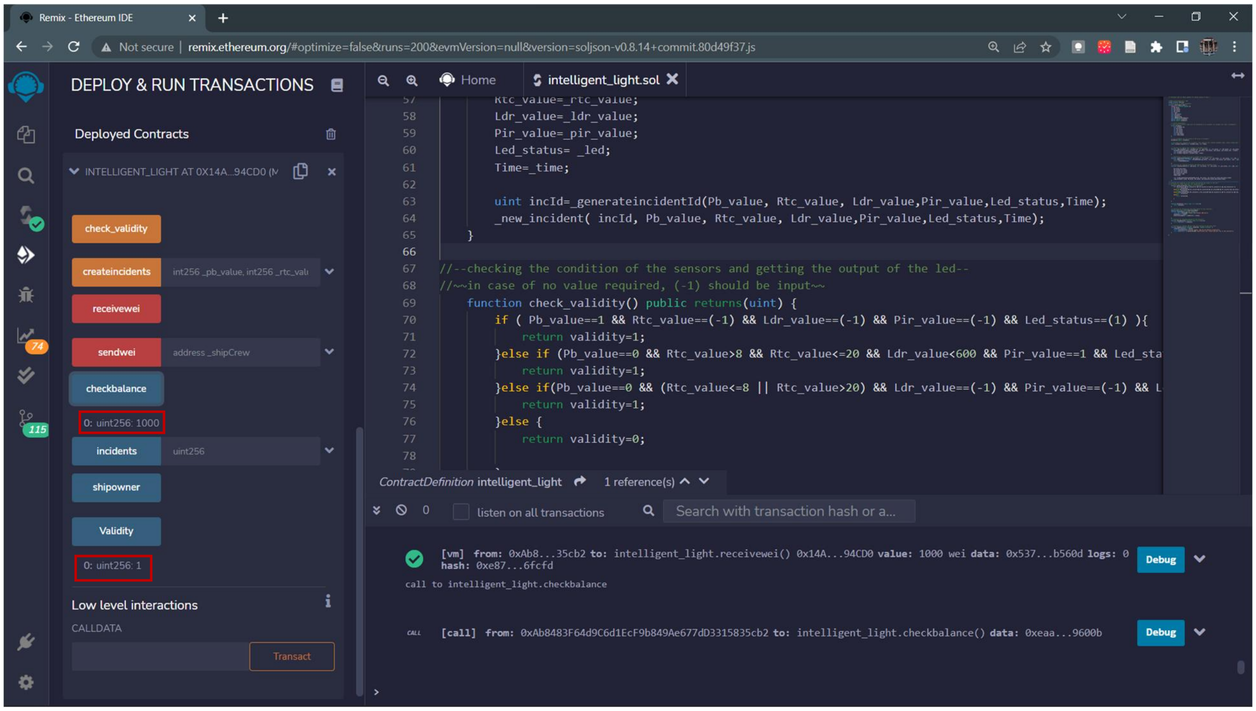

Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
Deployed Contracts
INTELLIGENT_LIGHT AT 0X14A...94CD0 (M
check_validity
createincidents
int256 _pb_value, int256 _rtc_valu
receivewei
sendwei
address _shipCrew
checkbalance
incidents
uint256
shipowner
Validity
74
115
Low level interactions
CALLDATA
Transact
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
Debug

0: uint256: 1

```
        Rtc_value=_rtc_value;
        Ldr_value=_ldr_value;
        Pir_value=_pir_value;
        Led_status= _led;
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

```

```
[vm]  from: 0xAb8...35cb2 to: intelligent_light.receivewei() 0x14A...94CD0 value: 500 wei data: 0x537...b560d logs: 0
hash: 0xd5c...247f9
transact to intelligent_light.receivewei pending ...

[vm]  from: 0xAb8...35cb2 to: intelligent_light.receivewei() 0x14A...94CD0 value: 1000 wei data: 0x537...b560d logs: 0
hash: 0xe87...6fcfd
```

h) **Scenario 8:** Balance Check



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js

checkbalance: 0: uint256: 1000

Validity: 0: uint256: 1

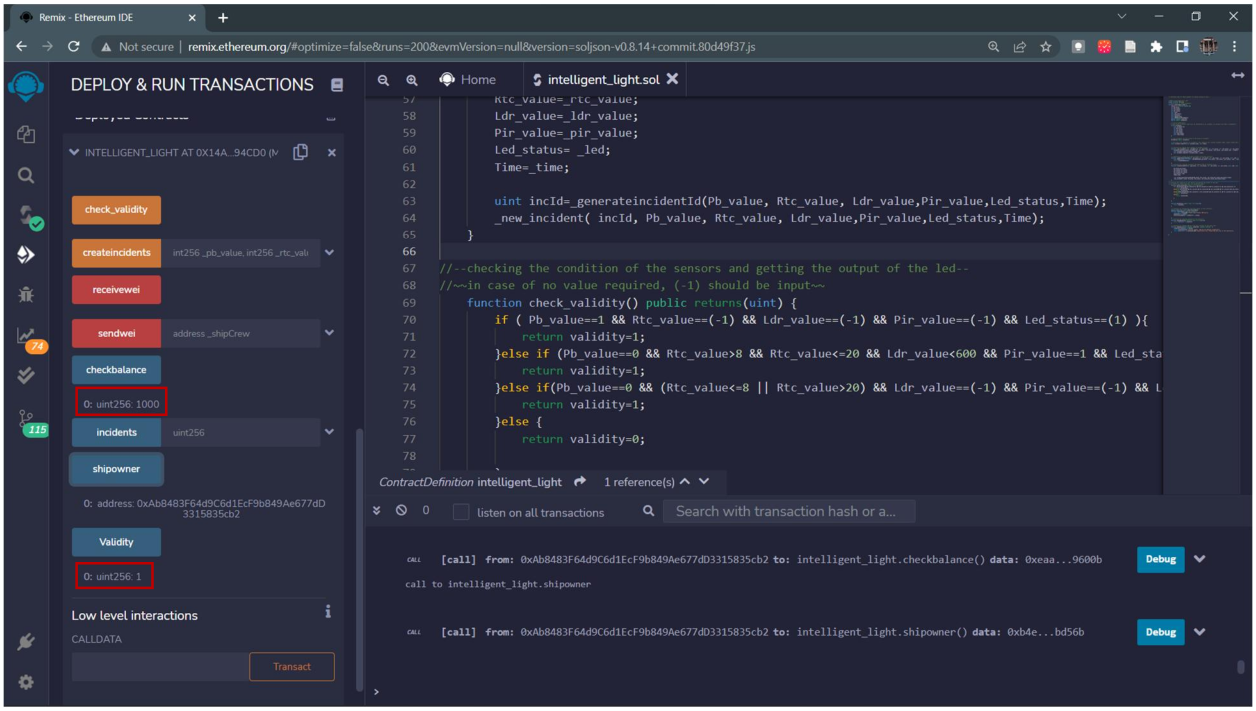
```
        Rtc_value=_rtc_value;
        Ldr_value=_ldr_value;
        Pir_value=_pir_value;
        Led_status= _led;
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

```

ContractDefinition intelligent_light 1 reference(s)

listen on all transactions | Search with transaction hash or a...

```
[vm] from: 0xAb8...35cb2 to: intelligent_light.receivewei() 0x14A...94CD0 value: 1000 wei data: 0x537...b560d logs: 0
hash: 0xe87...6fcfd
call to intelligent_light.checkbalance

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.checkbalance() data: 0xeaa...9600b
```

i) **Scenario 9:** Wei Sender Address Check



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js

checkbalance: 0: uint256: 1000

shipowner: 0: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

Validity: 0: uint256: 1

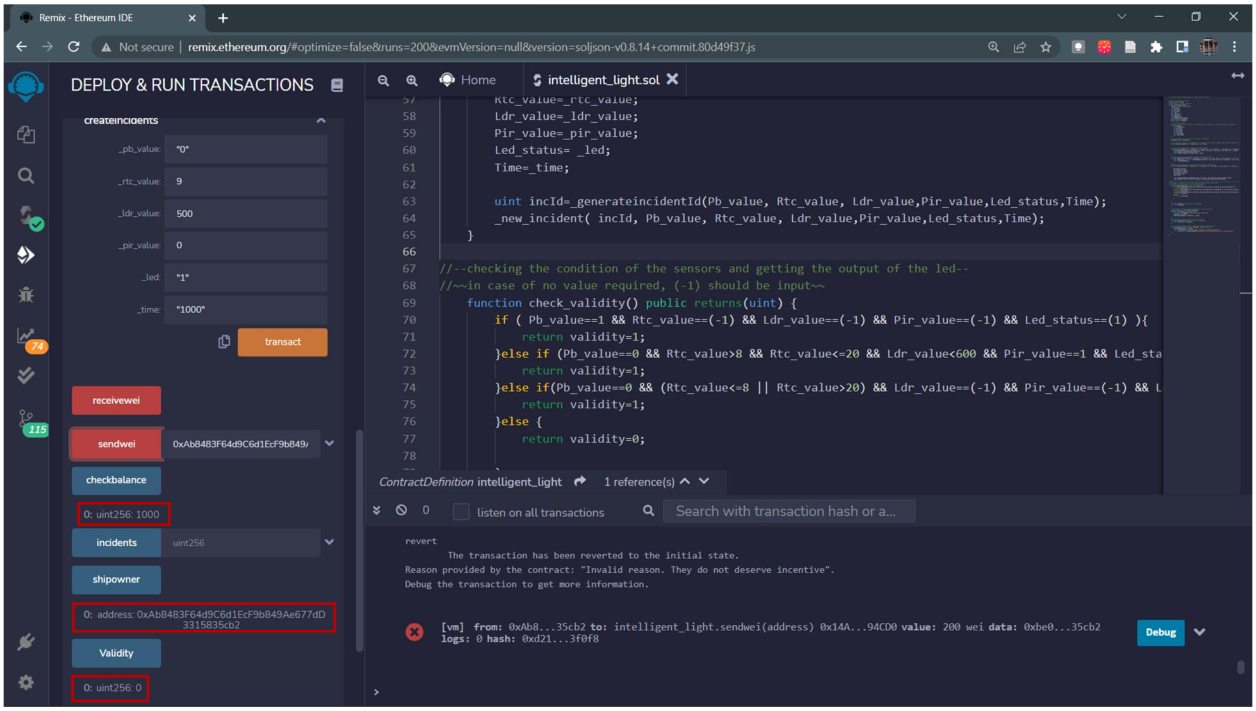
```
        Rtc_value=_rtc_value;
        Ldr_value=_ldr_value;
        Pir_value=_pir_value;
        Led_status= _led;
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;

```

ContractDefinition intelligent_light 1 reference(s)

listen on all transactions | Search with transaction hash or a...

```
CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.checkbalance() data: 0xeaa...9600b
call to intelligent_light.shipowner

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.shipowner() data: 0xb4e...bd56b
```

j) **Scenario 10:** Reason: Invalid, Sending Wei: Failed



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js

DEPLOY & RUN TRANSACTIONS

createincidents

_pb_value: "0"
_rtc_value: 9
_ldr_value: 500
_pir_value: 0
_led: "1"
_time: "1000"

checkbalance: 0: uint256: 1000

shipowner: 0: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

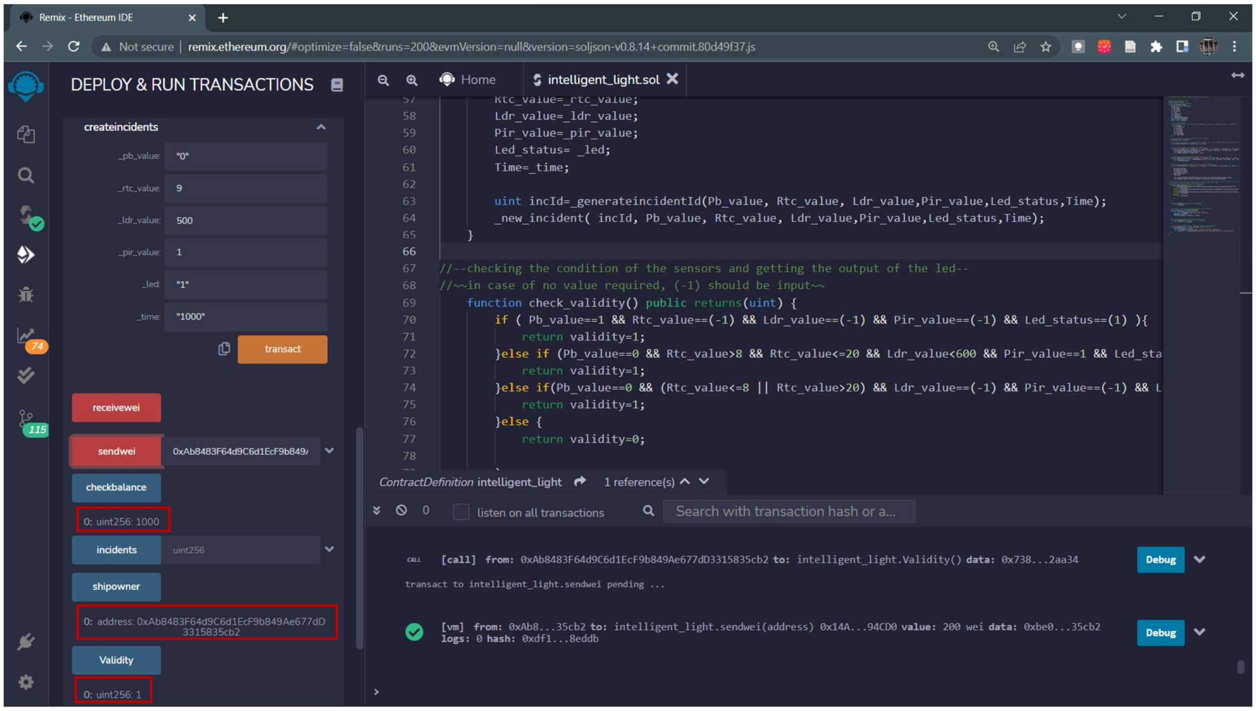
Validity: 0: uint256: 0

```
        Rtc_value=_rtc_value;
        Ldr_value=_ldr_value;
        Pir_value=_pir_value;
        Led_status= _led;
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;
```

```
revert
	The transaction has been reverted to the initial state.
Reason provided by the contract: "Invalid reason. They do not deserve incentive".
Debug the transaction to get more information.

[vm] from: 0xAb8...35cb2 to: intelligent_light.sendwei(address) 0x14A...94CD0 value: 200 wei data: 0xbe0...35cb2
logs: 0 hash: 0xd21...3f0f8
```

Remix - Ethereum IDE
Not secure
Home
intelligent_light.sol
transact
receivewei
sendwei
0xAb8483F64d9C6d1EcF9b849/
incidents
uint256
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
74
115

k) **Scenario 10:** Reason**:** Valid, Sending Wei: Executed



remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js

DEPLOY & RUN TRANSACTIONS

createincidents

_pb_value: "0"
_rtc_value: 9
_ldr_value: 500
_pir_value: 1
_led: "1"
_time: "1000"

checkbalance: 0: uint256: 1000

shipowner: 0: address: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2

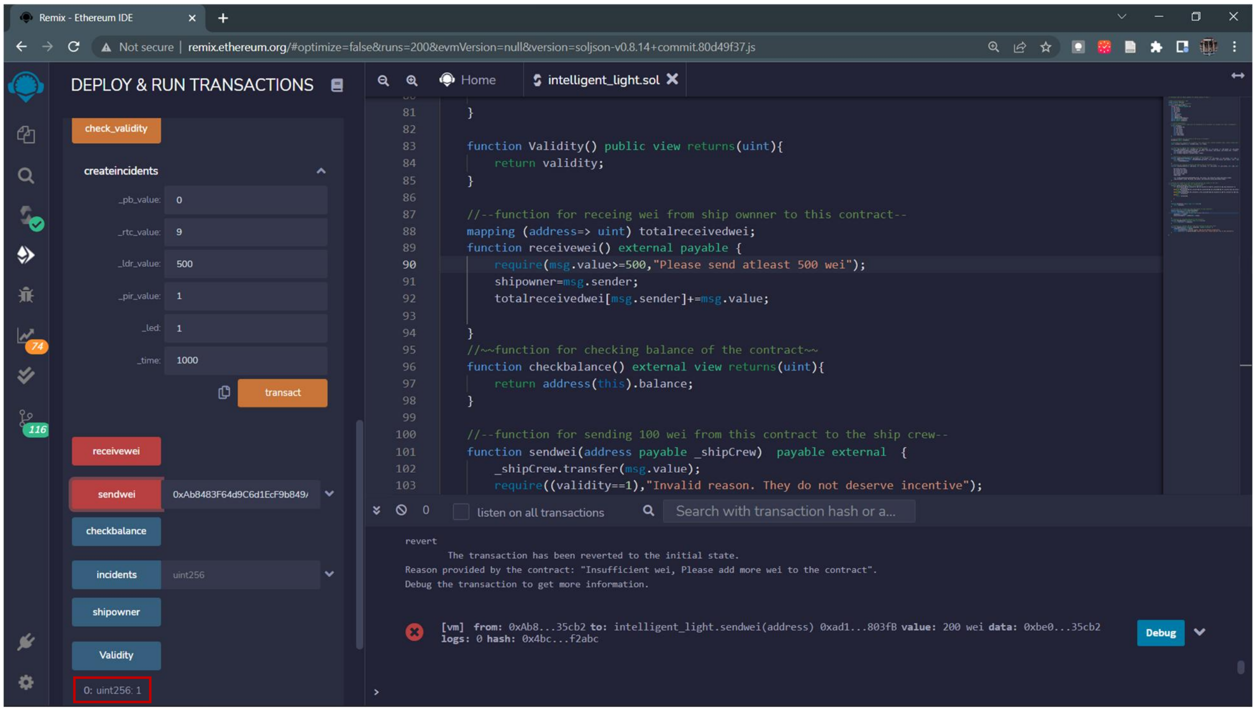
Validity: 0: uint256: 1

```
        Rtc_value=_rtc_value;
        Ldr_value=_ldr_value;
        Pir_value=_pir_value;
        Led_status= _led;
        Time=_time;

        uint incId=_generateincidentId(Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
        _new_incident( incId, Pb_value, Rtc_value, Ldr_value,Pir_value,Led_status,Time);
    }

//--checking the condition of the sensors and getting the output of the led--
//~~in case of no value required, (-1) should be input~~
    function check_validity() public returns(uint) {
        if ( Pb_value==1 && Rtc_value==(-1) && Ldr_value==(-1) && Pir_value==(-1) && Led_status==(1) ){
            return validity=1;
        }else if (Pb_value==0 && Rtc_value>8 && Rtc_value<=20 && Ldr_value<600 && Pir_value==1 && Led_sta
            return validity=1;
        }else if(Pb_value==0 && (Rtc_value<=8 || Rtc_value>20) && Ldr_value==(-1) && Pir_value==(-1) && L
            return validity=1;
        }else {
            return validity=0;
```

```
CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.Validity() data: 0x738...2aa34
transact to intelligent_light.sendwei pending ...

[vm] from: 0xAb8...35cb2 to: intelligent_light.sendwei(address) 0x14A...94CD0 value: 200 wei data: 0xbe0...35cb2
logs: 0 hash: 0xdf1...8eddb
```

Remix - Ethereum IDE
Not secure
Home
intelligent_light.sol
transact
receivewei
sendwei
0xAb8483F64d9C6d1EcF9b849/
incidents
uint256
ContractDefinition intelligent_light
1 reference(s)
listen on all transactions
Search with transaction hash or a...
Debug
74
115

l) **Scenario 11:** Balance: Insufficient, Sending Wei: Failed


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
74
116

check_validity

createincidents

_pb_value: 0
_rtc_value: 9
_ldr_value: 500
_pir_value: 1
_led: 1
_time: 1000

transact

receivewei

sendwei 0xAb8483F64d9C6d1EcF9b849/

checkbalance

incidents uint256

shipowner

Validity

0: uint256: 1

```
    }

    function Validity() public view returns(uint){
        return validity;
    }

    //--function for receing wei from ship ownner to this contract--
    mapping (address=> uint) totalreceivedwei;
    function receivewei() external payable {
        require(msg.value>=500,"Please send atleast 500 wei");
        shipowner=msg.sender;
        totalreceivedwei[msg.sender]+=msg.value;

    }
    //~~function for checking balance of the contract~~
    function checkbalance() external view returns(uint){
        return address(this).balance;
    }

    //--function for sending 100 wei from this contract to the ship crew--
    function sendwei(address payable _shipCrew)  payable external  {
        _shipCrew.transfer(msg.value);
        require((validity==1),"Invalid reason. They do not deserve incentive");
```

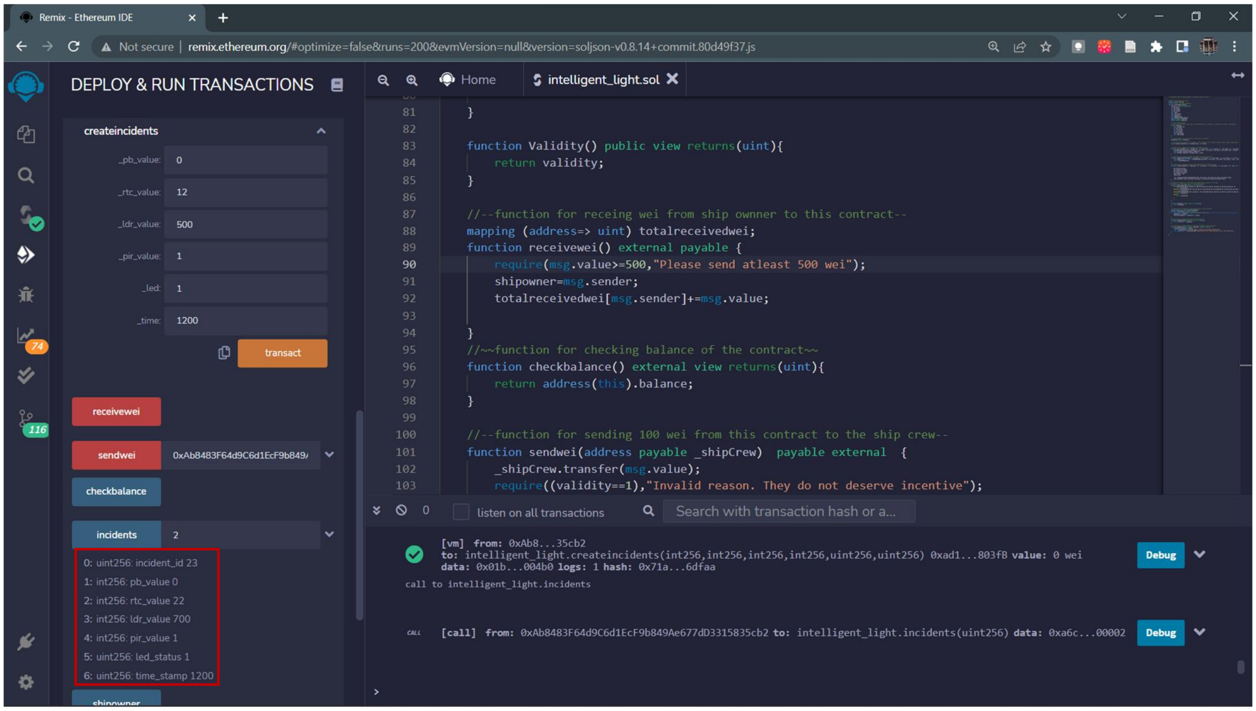
☐ listen on all transactions

Search with transaction hash or a...

```
revert
        The transaction has been reverted to the initial state.
Reason provided by the contract: "Insufficient wei, Please add more wei to the contract".
Debug the transaction to get more information.

[vm]  from: 0xAb8...35cb2 to: intelligent_light.sendwei(address) 0xad1...803fB value: 200 wei data: 0xbe0...35cb2
logs: 0 hash: 0x4bc...f2abc
```

Debug

m) **Scenario 12, 13:** Checking Log History


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
74
116

createincidents

_pb_value: 0
_rtc_value: 12
_ldr_value: 500
_pir_value: 1
_led: 1
_time: 1200

transact

receivewei

sendwei 0xAb8483F64d9C6d1EcF9b849/

checkbalance

incidents 2

0: uint256: incident_id 23
1: int256: pb_value 0
2: int256: rtc_value 22
3: int256: ldr_value 700
4: int256: pir_value 1
5: uint256: led_status 1
6: uint256: time_stamp 1200

shipowner

```
    }

    function Validity() public view returns(uint){
        return validity;
    }

    //--function for receing wei from ship ownner to this contract--
    mapping (address=> uint) totalreceivedwei;
    function receivewei() external payable {
        require(msg.value>=500,"Please send atleast 500 wei");
        shipowner=msg.sender;
        totalreceivedwei[msg.sender]+=msg.value;

    }
    //~~function for checking balance of the contract~~
    function checkbalance() external view returns(uint){
        return address(this).balance;
    }

    //--function for sending 100 wei from this contract to the ship crew--
    function sendwei(address payable _shipCrew)  payable external  {
        _shipCrew.transfer(msg.value);
        require((validity==1),"Invalid reason. They do not deserve incentive");
```

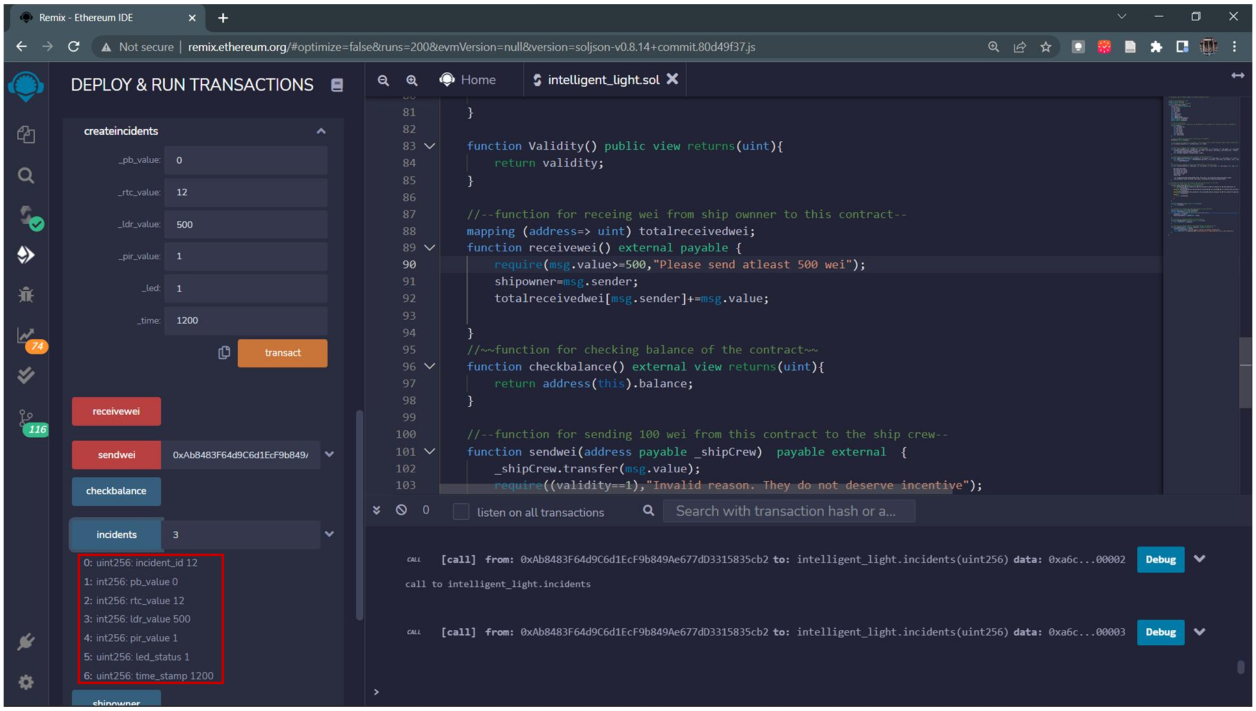
☐ listen on all transactions

Search with transaction hash or a...

```
[vm]  from: 0xAb8...35cb2
to: intelligent_light.createincidents(int256,int256,int256,int256,uint256,uint256) 0xad1...803fB value: 0 wei
data: 0x01b...004b0 logs: 1 hash: 0x71a...6dfaa
call to intelligent_light.incidents

CALL  [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.incidents(uint256) data: 0xa6c...00002
```

Debug

Debug


Remix - Ethereum IDE
Not secure | remix.ethereum.org/#optimize=false&runs=200&evmVersion=null&version=soljson-v0.8.14+commit.80d49f37.js
DEPLOY & RUN TRANSACTIONS
Home
intelligent_light.sol
74
116
transact
receivewei
sendwei
checkbalance
incidents
shipowner
listen on all transactions
Search with transaction hash or a...
Debug
Debug

createincidents

_pb_value: 0

_rtc_value: 12

_ldr_value: 500

_pir_value: 1

_led: 1

_time: 1200

sendwei 0xAb8483F64d9C6d1EcF9b849/

incidents 3

```
0: uint256: incident_id 12
1: int256: pb_value 0
2: int256: rtc_value 12
3: int256: ldr_value 500
4: int256: pir_value 1
5: uint256: led_status 1
6: uint256: time_stamp 1200
```

```
    }

    function Validity() public view returns(uint){
        return validity;
    }

    //--function for receing wei from ship ownner to this contract--
    mapping (address=> uint) totalreceivedwei;
    function receivewei() external payable {
        require(msg.value>=500,"Please send atleast 500 wei");
        shipowner=msg.sender;
        totalreceivedwei[msg.sender]+=msg.value;

    }
    //~~function for checking balance of the contract~~
    function checkbalance() external view returns(uint){
        return address(this).balance;
    }

    //--function for sending 100 wei from this contract to the ship crew--
    function sendwei(address payable _shipCrew)  payable external  {
        _shipCrew.transfer(msg.value);
        require((validity==1),"Invalid reason. They do not deserve incentive");
```

```
CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.incidents(uint256) data: 0xa6c...00002
call to intelligent_light.incidents

CALL [call] from: 0xAb8483F64d9C6d1EcF9b849Ae677dD3315835cb2 to: intelligent_light.incidents(uint256) data: 0xa6c...00003

>
```

7. Conclusion:

Blockchain is a contemporary and revolutionary technology. The scope of the smart contract will eliminate the necessity of intermediary, increase trust, and security and streamline various processes. Solidity is a popular computer language to write the smart contract. Our solidity code is working as expected. However, it can be further improved to perform more complex tasks. This assignment has instigated our interest and is a stepping stone in the world of blockchain. While writing this code we received support from Professor Tiago M. Fernández Caramés and Dr. Paula Fraga Lamas. Various resources from the internet also helped us to modify and debug the code.