International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 11, November 2021)
Manuscript Received: 02 October 2021, Received in Revised form: 04 November 2021, Accepted: 10 November 2021 DOI: 10.46338/ijetae1121_19

Digital System Design for Traffic Light Controller System: A Systematic Approach

To-Nhi Ho. T[1,3*], Giao N. Pham[2*], Quang-Hung Nguyen[1,3], Binh A. Nguyen[4], Ngoc T. Le[4], and Hoanh-Su Le[1,3**]

[1]*University. of Economics and Law, Ho Chi Minh City, Vietnam*
[2]*Dept. of Computing Fundamentals,* [4]*ICT Department, FPT University, Hanoi, Vietnam*
[3]*Vietnam National University, Ho Chi Minh City, Vietnam*

**Co-first author, **Corresponding author*

***Abstract*—In this paper, we are going to present the finite state machine, how to implement it via hardware description language (HDL), and how to use it in a real application. At first, the specification and requirements of traffic light controller are stated. Then, the system architecture based on finite state machine (FSM) are conducted. Finally, the way of using HDL as well as the test-bench simulation are given in detail.**

***Keywords*-- Digital system design, System on chip, Finite State Machine, Digital Design Education, Smart Classroom.**

I. INTRODUCTION

Nowadays, everyone believes that excessive use of modern technologies such as computers, smartphones, cars, telecommunication systems, cloud computing, is affecting the human living [1]. And the technological equipment is changing day by day thanks to the contribution of scientist and engineers on their effort on digital system designs [2].

Finite state machine or finite state automation, or simply state machine, is a mathematical model of computation, an abstract machine that can be in exactly one of a finite number of states at any given time. In digital circuit, an FSM could be implemented by using programmable logic devices, a programmable logic controller, logic gates and flip flops or relays [3]. Thanks for the contribution of hardware description languages such as Verilog HDL, VHDL, we can make the synthesizable programs like software coding to make the digital circuits. As the consequence, the roads to real application or system are shorter [4]. In addition, the education of digital application design is more convenient with the simulation and implementation test-bed in the smart classroom environment [5-6].

In this paper, by considering the application of FSM (in section II) on traffic light controller, we will steps by steps (in section III) give the problem statements, algorithms designs, Hardware description language implementation, and test-bench simulation.

II. FINITE STATE MACHINE FUNDAMENTALS

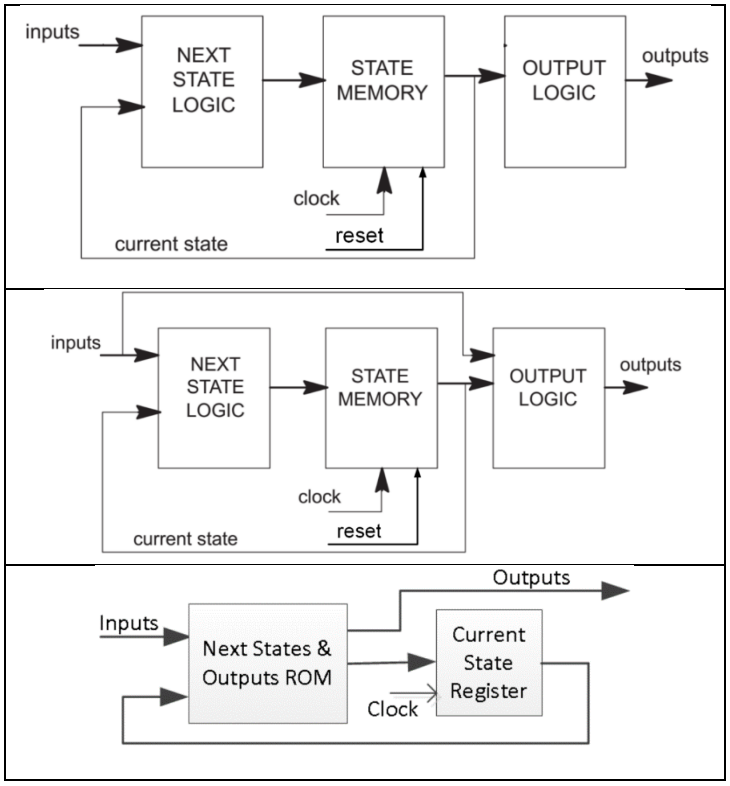


Figure I. Famous Architectures Of FSM System

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 11, November 2021)

Finite State Machine (FSM) is a basic but very important digital model, which is popularly used in digital designs due to its advantages on system debug, easy operational tracking. FIGURE 1 shows two FSM architectures: FSM Moore and FSM Mealy, the first part is for Moore and the second & third parts are for Mealy. The differences between Mealy and Moore models are the output values are determined both by its current state and the current inputs.

III. Digital System Design For Traffic Light Controller System

A. Problem statement

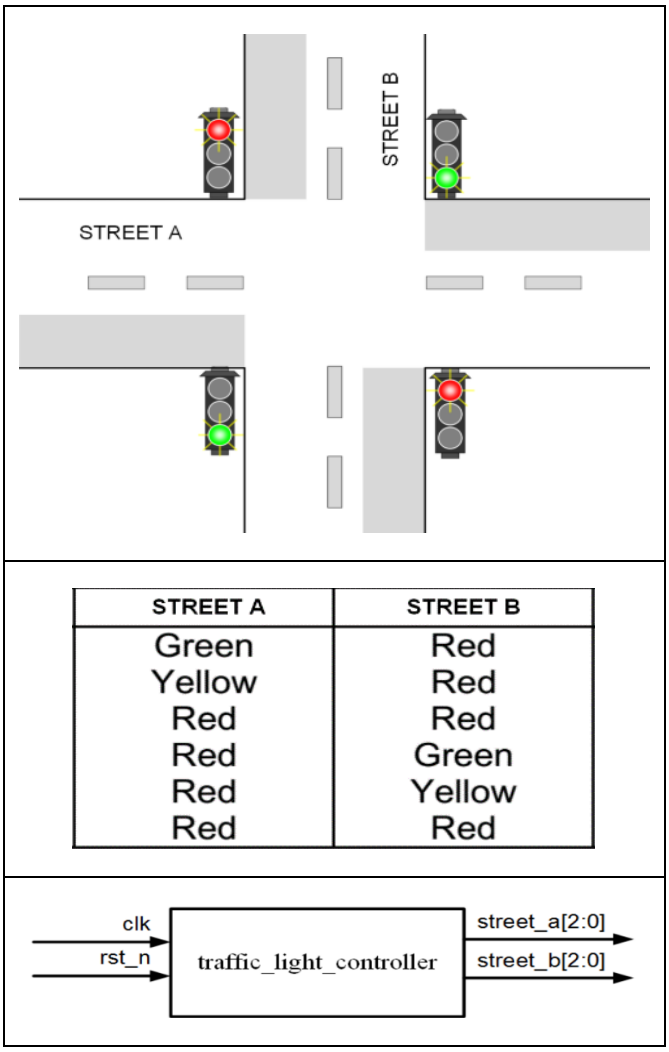
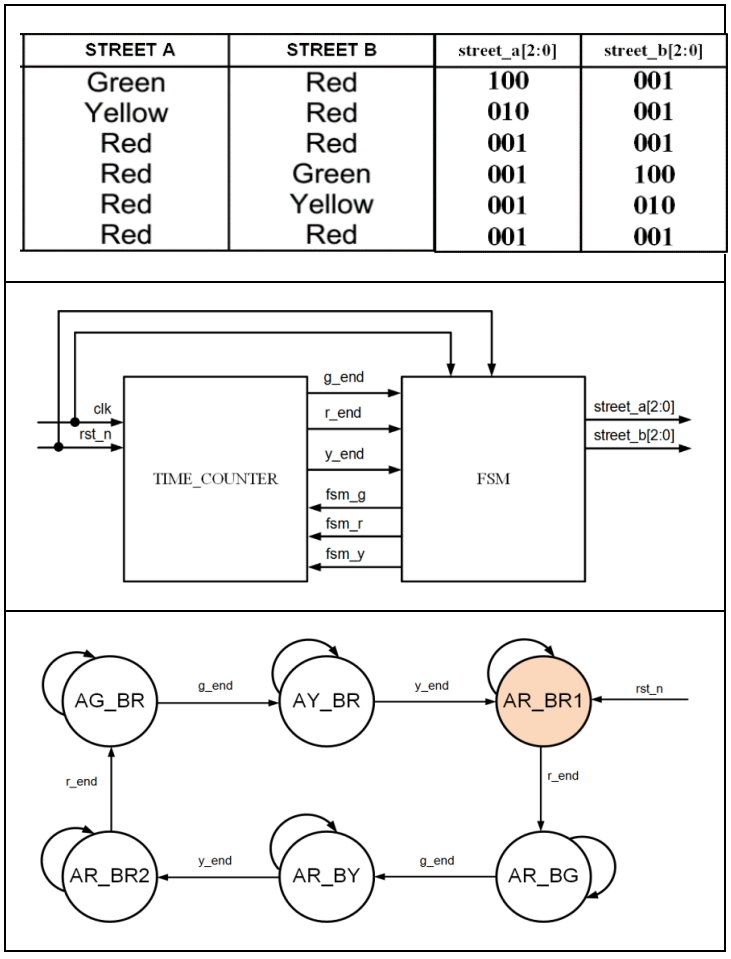


Figure III. Requirements For Traffic Light System

We will consider the traffic light in a crossroad where the considered system includes two traffic lights for street A and street B. Each light has three types of light signals: RED, YELLOW, and GREEN. The light state is summarized as the table in FIGURE II. In addition, the overall design of proposed traffic light controller is also given in FIGURE II where *clk* is the main clock system assumed 1Hz frequency; *rst_n* is the reset signal to initialize our system; *street_a* and *street_b* are the output signals for light control.

B. FSM modules analysis



International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 11, November 2021)

State	street_a[2:0]	street_b[2:0]	fsm_g	fsm_y	fsm_r
AG_BR	100	001	1	0	0
AY_BR	010	001	0	1	0
AR_BR1	001	001	0	0	1
AR_BG	001	100	1	0	0
AR_BY	001	010	0	1	0
AR_BR2	001	001	0	0	1

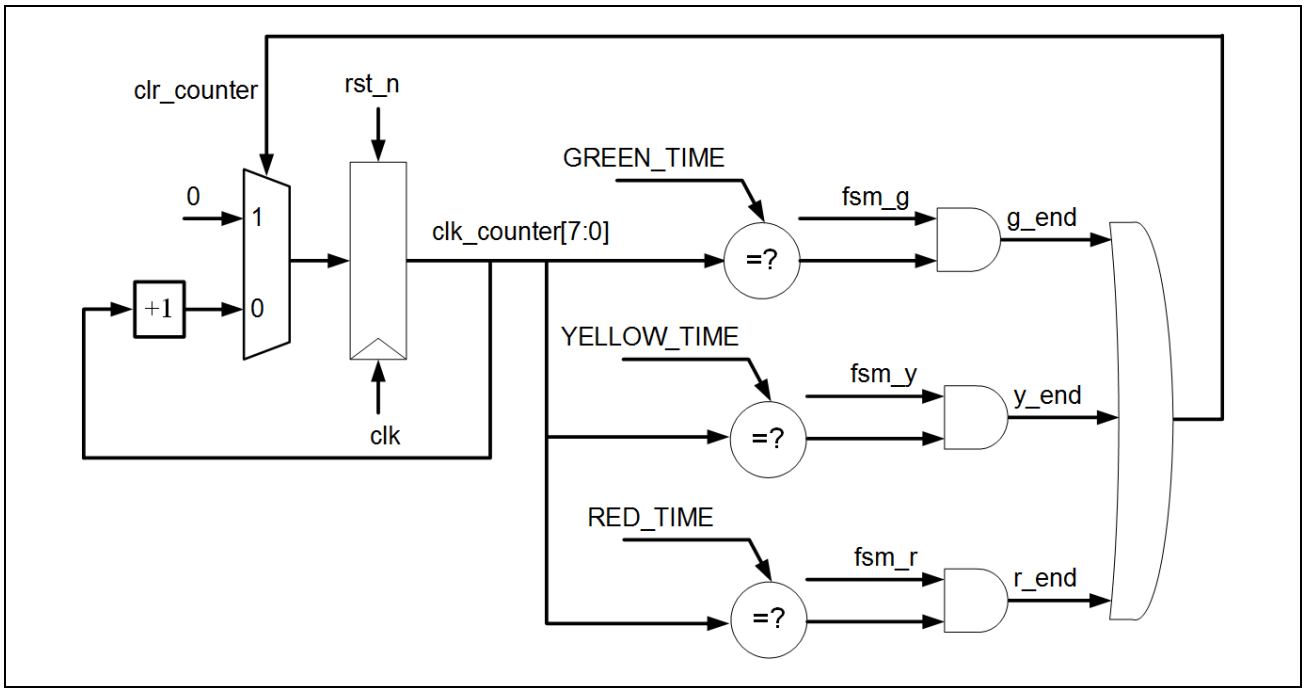
Figure IV. Expected output signals with corresponding light states

Based on the requirement analysis in section A, the expected outputs of proposed controller for *street_a* and *street_b* are given in the table of FIGURE III. And we divide the controller into two smaller units: TIME_COUNTER and FSM.

TIME_COUNTER is for timing, which count the pre-defined clocks such as 29 clocks for GREEN_TIME, 4 clocks for YELLOW_TIME, and 2 clocks for RED_TIME. FSM shows the transition between finite states, we will have six states: AG_BR (*street_a* is red and *street_b* is red), AY_BR (*street_a* is yellow and *street_b* is red), and so on. The transitions between those states, and the condition of transitions are also given, and a note for an assumption is that the reset state is AR_BR1 (both lights are red).

C. System implementations via HDL

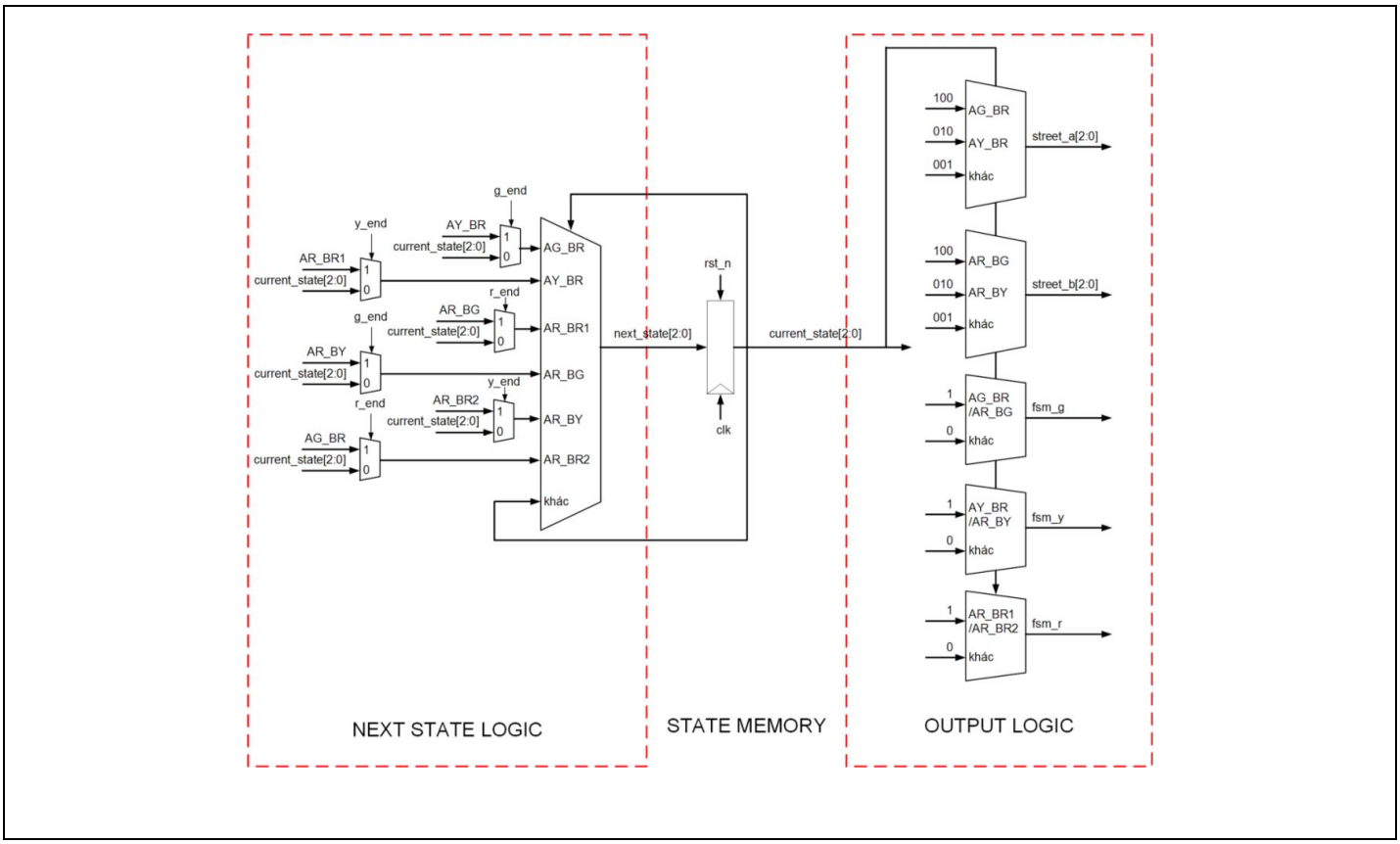
Our system will be implemented based on Verilog hardware description language for synthesizable and the design is tested in FPGA board. The whole designs include TIME_COUNTER unit (as FIGURE IV), FSM unit (as FIGURE V), TOP system module, and a test-bench simulation as FIGURE VI.



International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 11, November 2021)

```
always @ (posedge clk) begin
  if (~rst_n) clk_counter[7:0] <= 8'd0;
  else if (clr_counter) clk_counter[7:0] <= 8'd0;
  else clk_counter[7:0] <= clk_counter[7:0] + 1'b1;
end
//Compare the end time
assign g_end = fsm_g & (clk_counter[7:0] == GREEN_TIME);
assign y_end = fsm_y & (clk_counter[7:0] == YELLOW_TIME);
assign r_end = fsm_r & (clk_counter[7:0] == RED_TIME);
//
assign clr_counter = g_end | y_end | r_end;
```

Figure IV: Implementation of time_counter unit using Verilog hardware description language.



International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 11, November 2021)

```
//STATE code
localparam AG_BR  = 3'd0;
localparam AY_BR  = 3'd1;
localparam AR_BR1 = 3'd2;
localparam AR_BG  = 3'd3;
localparam AR_BY  = 3'd4;
localparam AR_BR2 = 3'd5;
```

```
//STATE MEMORY
always @ (posedge clk) begin
  if (~rst_n) current_state[2:0] <= AR_BR1;
  else current_state[2:0] <= next_state[2:0];
end
```

```
//Next state logic
always @ (*) begin
  case (current_state[2:0])
    AG_BR: begin
      if (g_end) next_state[2:0] = AY_BR;
      else next_state[2:0] = current_state[2:0];
    end
    AY_BR: begin
      if (y_end) next_state[2:0] = AR_BR1;
      else next_state[2:0] = current_state[2:0];
    end
    AR_BR1: begin
      if (r_end) next_state[2:0] = AR_BG;
      else next_state[2:0] = current_state[2:0];
    end
    AR_BG: begin
      if (g_end) next_state[2:0] = AR_BY;
      else next_state[2:0] = current_state[2:0];
    end
    AR_BY: begin
      if (y_end) next_state[2:0] = AR_BR2;
      else next_state[2:0] = current_state[2:0];
    end
    AR_BR2: begin
      if (r_end) next_state[2:0] = AG_BR;
      else next_state[2:0] = current_state[2:0];
    end
    default: next_state[2:0] = current_state[2:0];
  endcase
end
```

```
//Output logic
always @ (*) begin
  case (current_state[2:0])
    AG_BR: street_a[2:0] = 3'b100;
    AY_BR: street_a[2:0] = 3'b010;
    default: street_a[2:0] = 3'b001;
  endcase
end
//
always @ (*) begin
  case (current_state[2:0])
    AR_BG: street_b[2:0] = 3'b100;
    AR_BY: street_b[2:0] = 3'b010;
    default: street_b[2:0] = 3'b001;
  endcase
end
//
assign fsm_g = (current_state[2:0] == AG_BR) | (current_state[2:0] == AR_BG);
assign fsm_y = (current_state[2:0] == AY_BR) | (current_state[2:0] == AR_BY);
assign fsm_r = (current_state[2:0] == AR_BR1) | (current_state[2:0] == AR_BR2);
```

Figure V: Implementation Of Finite-State-Machine Unit Using Verilog Hardware Description Language.

International Journal of Emerging Technology and Advanced Engineering
Website: www.ijetae.com (E-ISSN 2250-2459, Scopus Indexed, ISO 9001:2008 Certified Journal, Volume 11, Issue 11, November 2021)

```
time_counter cnt (/*AUTOINST*/
                 // Outputs
                 .g_end          (g_end),
                 .y_end          (y_end),
                 .r_end          (r_end),
                 // Inputs
                 .clk            (clk),
                 .rst_n          (rst_n),
                 .fsm_g          (fsm_g),
                 .fsm_r          (fsm_r),
                 .fsm_y          (fsm_y));

fsm fsm (/*AUTOINST*/
         // Outputs
         .street_a           (street_a[2:0]),
         .street_b           (street_b[2:0]),
         .fsm_g              (fsm_g),
         .fsm_y              (fsm_y),
         .fsm_r              (fsm_r),
         // Inputs
         .clk                (clk),
         .rst_n              (rst_n),
         .g_end              (g_end),
         .y_end              (y_end),
         .r_end              (r_end));
```

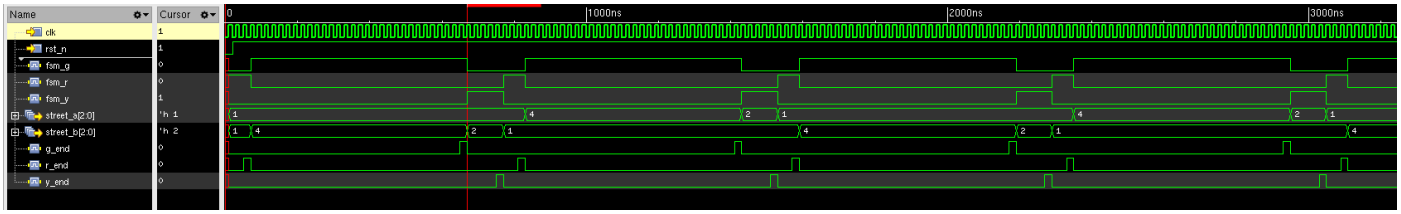


Figure VI: Top Simulation Of Traffic Light Controller Using Verilog Hardware Description Language.

IV. CONCLUSION

In this paper, the fundamental concepts of finite state machine have been introduced. In addition, we have analyzed an outstanding project named traffic light controller steps by steps from requirement collections, digital system design, RTL coding, Test-Bench simulation. We believe that in digital design education, it is convenient and for simulation and implement test-bench for students to practice in a smart classroom or a laboratory. We also hope that digital design engineer will have a good reference for their job.

In the future, will implement this application in a smart classroom environment to help students experiment digital design, smart transportation simulation projects. All the source codes are open and will be shared as requirement.

Acknowledgement

This research is funded by Vietnam National University Ho Chi Minh City (VNU-HCM) under grant number: C2020-34-01.

REFERENCES

[1] Nguyen, D. M. et. al. “New constructions of quantum stabilizer codes based on difference sets”. Symmetry 10 (11), 655, 2018.

[2] Nguyen, D. M. et. al. “A novel construction for quantum stabilizer codes based on binary formalism”. International Journal of Modern Physics B 34 (8), 2050059, 2020.

[3] Giao, N. P. et. al. “Palm rejection algorithms on touch screen communication system”. World Journal of Advanced Engineering Technology and Sciences 2021, 02(02), 052-057

[4] Giao, N. P. et. al. “Implementation of Differential Sensing Scheme to Remove Panel Noise in Touchscreen Controller”. International Journal of Emerging Technology and Advanced Engineering, 2021, 11(10), 104-108

[5] Cebrian, Gisela, Ramon Palau, and Jordi Mogas. "The smart classroom as a means to the development of ESD methodologies". Sustainability 12.7, 2020.

[6] Zhang, Mingbao, and Xiang Li. "Design of Smart Classroom System Based on Internet of Things Technology and Smart Classroom". Mobile Information System, 2021.