Middle East Technical University **Ceng336**
Computer Engineering Department **Spring 2018-2019**

(Individual) Take Home Exam-1

Revision 1.0

Due: Mar 8, 2019, 23:55 hrs

Submission: ***via ODTUCLASS***

The purpose of this assignment is to familiarize you with **basic input/output operations on MPLAB X IDE simulation** environment.

Any clarifications and revisions for the assignment will be posted to the discussion group on ODTUCLASS.

This is not a group exam; you must solve it **individually.**

Your mission is to implement the below scenario using the MPLAB X IDE simulation environment. You can also program the MCDEV boards to observe actual behavior.

Scenario

You are expected to use 16 **LEDs**, and 3 **push buttons to** draw a light pattern**.** Your program should have three states. First of all, you must turn on the first four LEDs of all specified ports for 2 seconds and then turn off all the LEDs for 1 second. In state 1, the LEDs are turned on one by one. At this time if the pause button (RA4) is pressed and released, LEDs must be stopped until the other push buttons (RE3 or RE4) are pressed and released. In State 2, your program should wait until pressing a push button (RE3 or RE4). This action will trigger your program. If the push button RE3 is pressed and released, then your program will go to State 1, on the other hand, if the push button RE4 is pressed and released, it will go to State 3. In State 3, pressing the push button RE4 will lead your program to turn off the LEDs one by one.

Specifications

Five ports will be used in this assignment: **PORTA**, **PORTB**, **PORTC, PORTD, and PORTE**. LEDs are connected to the following pins **RA[0-3], RB[0-3], RC [0-3]** and **RD [0-3].** The push buttons are **RA4, RE3, and RE4 which** must be configured as **digital inputs,** please look at the relevant section in datasheet.

The default state of the program is State1. At the very beginning of the program, you must turn on **the first four LEDs of all the ports** (16 pins in total) for 2 seconds and then turn off **all LEDs** for 1 second as it can be seen in Figure 4-5. Afterwards, the LEDs are turned on one by one start from **RA0**. The light pattern will be in the following order: **RA0, RA1, RA2, RA3, RB0, RB1, RB2, RB3, RC0, RC1, RC2, RC3, RD0, RD1, RD2, RD3**. State 1 is illustrated with nine sample pictures in Figure 6. If the **RA4** button is pressed and released, then turning on/off the LEDs stop, and the program waits for an **RE3 or RE4** button action.

In State 2, Whenever **RA4** is pressed and released, the program waits for a button action. The RE3 button is pressed and released, the LEDs start to be turned on from **the LED which is turned on the last time** as it can be seen in Figure 7. When all LEDs are turned on, the program should wait for the **RA4** button action. The RE4 button is pressed and released, the LEDs start to be turned off from the LED which is **turned on the last time**. State 3 is illustrated with three sample pictures in

Figure 8. When all LEDs are turned off, the program should wait for the **RA4** button action. In case of all LEDs are turned on or off, the program should wait only the **RA4** button action. Then your program will go to State2, in other words, your program will wait a push button action (RE3/RE4). **LED switching during the movement must be done in 500 ms (+/- 50 ms)**.

The naming of pins is <R>+"PORT NAME"+"BIT ORDER". R represents PORT. Here, ports names are A, B, C, and D. And the range of the BIT ORDER is between 0 and 7.

Implementation

You are expected to simulate the above scenario in the **MPLAB X IDE simulator**. You can track the status of the LEDs and buttons, by using **RA0, RA1, RA2, RA3, RB0, RB1, RB2, RB3, RC0, RC1, RC2, RC3, RD0, RD1, RD2, RD3, RA4, RE3 and RE4** pins of the simulator. The output of the pins can be seen by using the **I/O pins menu** which is available under the **Window --> Simulator** menu. A screen shot from the I/O pins menu of the simulator is given in *Figure 1*. There you can find states of some pins that are representing the LEDs and buttons. You can also simulate push buttons by using the **Stimulus** menu. This feature is available under the **Window --> Simulator** menu. You can control voltage levels of the buttons as High or Low by clicking the **RA4**, and **RE3-4** pin button which is can be seen in *Figure 2*. You can add the pin corresponded with push button by using **the I/O pins menu** to follow the changes on this pin. **Note that** in *Figure 1* and *Figure 2*, some pins are not relevant to this assignment and only given as an example.

Coding Rules:

- You will code your program using PIC assembly language.
- Your program should be written for **PIC18F8722** working at **40 MHz.**
- When the push button is pressed, then RA4/RE[3-4] pin goes high, and when we release the push button, the RA4/RE[3-4] pin goes low.
- When you are writing your codes, please look at the relevant lecture notes and recitation documents. In particular, you are expected to apply the **round robin approach** to your code, so your program will execute in an infinite loop, and there will be **your own tasks for your states** in this loop at the main scope.
- Be precise for constructing time delays (intervals). You may need to count the number of instructions for those sections.
- Each button action should be active after **the button pressed and released (not only pressed)**.

Hand in Instructions

- You should submit your code as a single file named as `the_1_##.asm` through ODTUCLASS where ## represents your seven-digit student number.

Hints

- You can use "Stopwatch" tool of the simulator in MPLAB X IDE to measure the time spent by a code segment. You can reach this tool from "`Windows -> Debugging -> Stopwatch`" menu. But, before starting simulator you must configure your clock speed to 10 MHz from "`Project properties -> Simulator -> Instruction Frequency`" (Since 40 MHz main clock frequency is divided by 4 inside microcontroller and one instruction cycle is equal to 4 cycle of 40 MHz clock signal, you are setting "Instruction Frequency" to 10 MHz). You can reach the project properties by right clicking to the project name in the left side "Projects" panel and selecting properties. By **putting breakpoints** on the lines between which you want to measure the amount of time spent by that code segment and running the code

between these two breakpoints, you can see the time spent in stopwatch window.

- You will use **RA4 pin** as a **digital input** by configuring a **special register**, please check the data sheet for this configuration.
- You can also see the states of your ports, variables and pins by using the logical analyzer, SFR and variables menu under the window menu.

Grading

Your codes will be evaluated on the MPLAB X simulation environment. However, if you obey the specified rules, the program can directly be executed on the boards without any problem. You can test your code also in your MCDEV Kit. Please, look at the related switch configuration which is provide you in the exam files. This an individual exam but the group members can execute their codes with the dedicated MCDEV Kit to this group.

Resources

- PIC 18F8722 Datasheet
- ODTUCLASS course web page
- The **MPLAB X IDE** will be used. You can download by using the following link:

http://www.microchip.com/pagehandler/en-us/family/mplabx/home.html

Stimulus | I/O Pins | Output | SFRs | File Registers | Watches | Variables | Call Stack | Breakpoints

Pin	Mode	Value	Owner or *Mapping*
RD3	Dout	1	**RD3**/AD3/PSP3
RD2	Dout	1	**RD2**/AD2/PSP2
RD1	Dout	1	**RD1**/AD1/PSP1
RD0	Dout	1	**RD0**/AD0/PSP0
RC4	Din	0	**RC4**/SDI1/SDA1
RC3	Dout	1	**RC3**/SCK1/SCL1
RC2	Dout	0	**RC2**/ECCP1/P1A
RC1	Dout	0	**RC1**/T1OSI/ECCP21/P2A1
RC0	Dout	1	**RC0**/T1OSO/T13CKI
RB3	Dout	1	**RB3**/INT3/ECCP20/P2A0
RB2	Dout	0	**RB2**/INT2
RB1	Dout	0	**RB1**/INT1
RB0	Dout	1	**RB0**/INT0/FLT0
RA4	Din	0	**RA4**/T0CKI
RA3	Dout	1	**RA3**/AN3/VREF+
RA2	Dout	1	**RA2**/AN2/VREF-
RA1	Dout	1	**RA1**/AN1
RA0	Dout	1	**RA0**/AN0
<New Pin>			

Figure 1: State of various LED and button pins shown in I/O pins window.

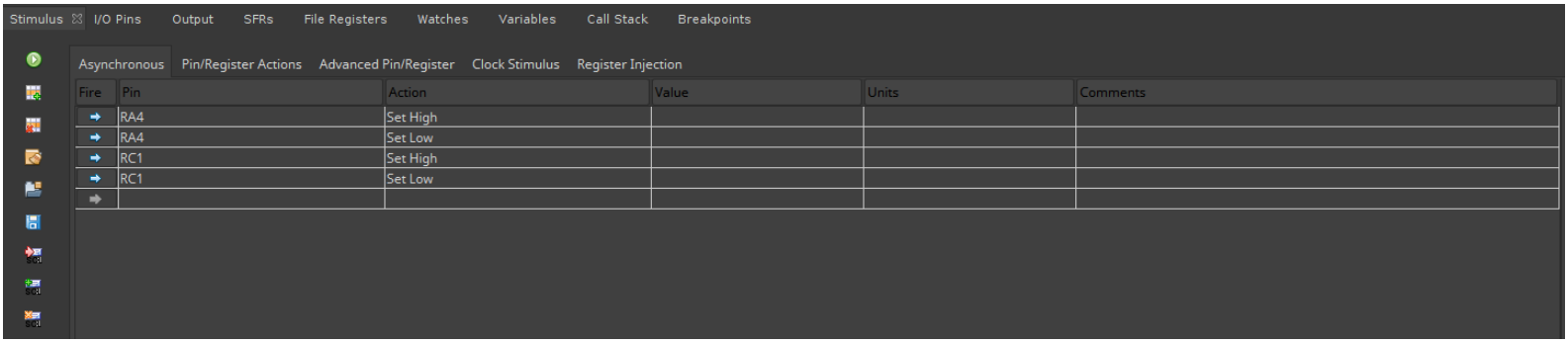


Figure 2: Example button (RA4 & RC1) control using the stimulus window.

Program Snapshots

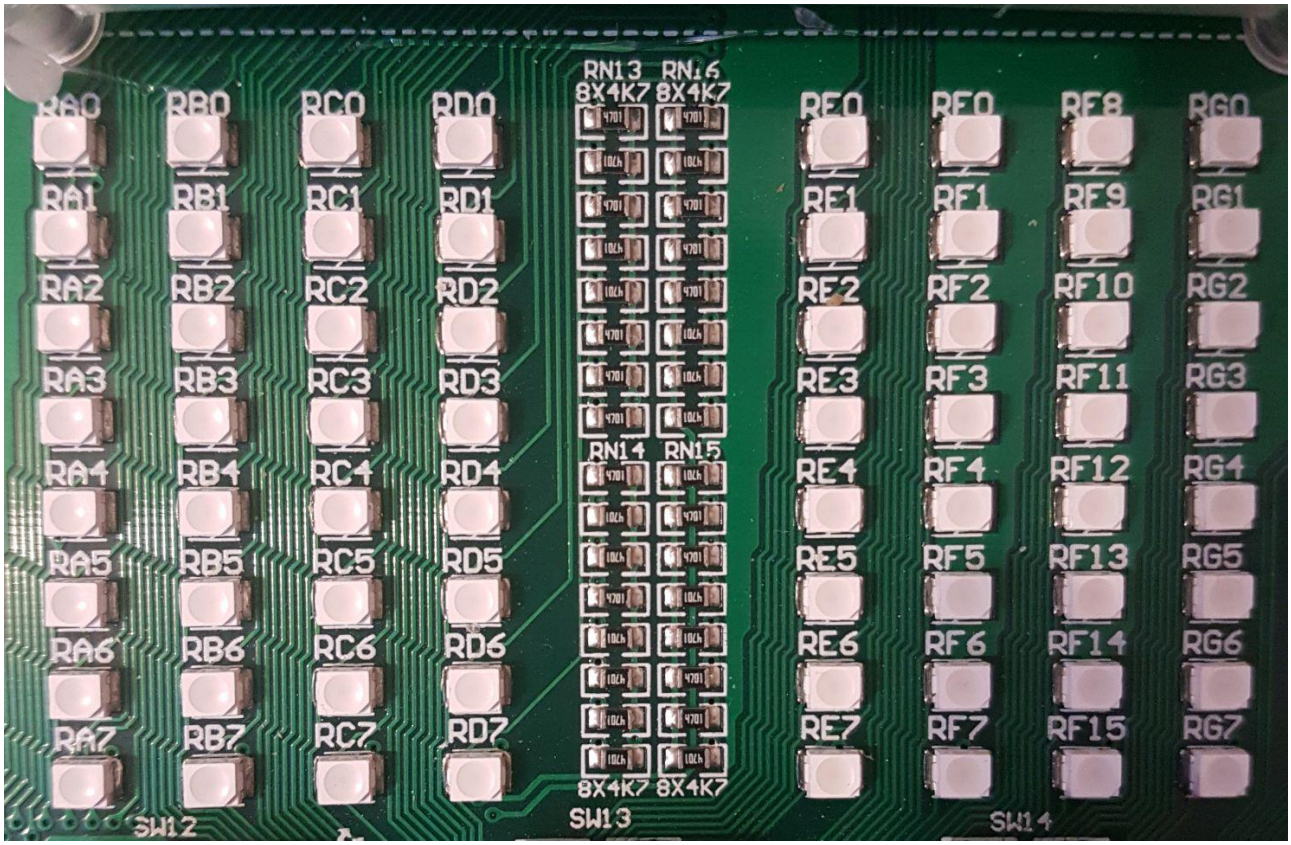


Figure 3. Board is off

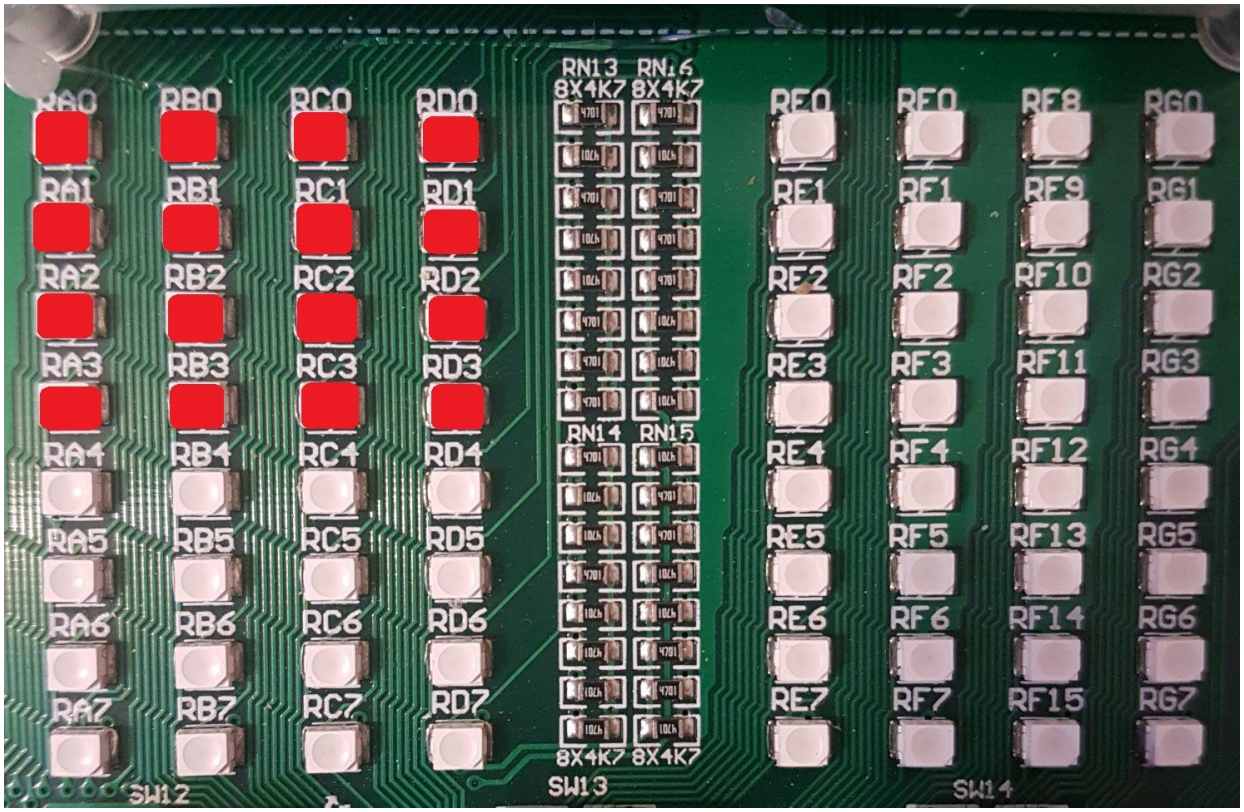


Figure 4. Initial Configuration for 2 seconds

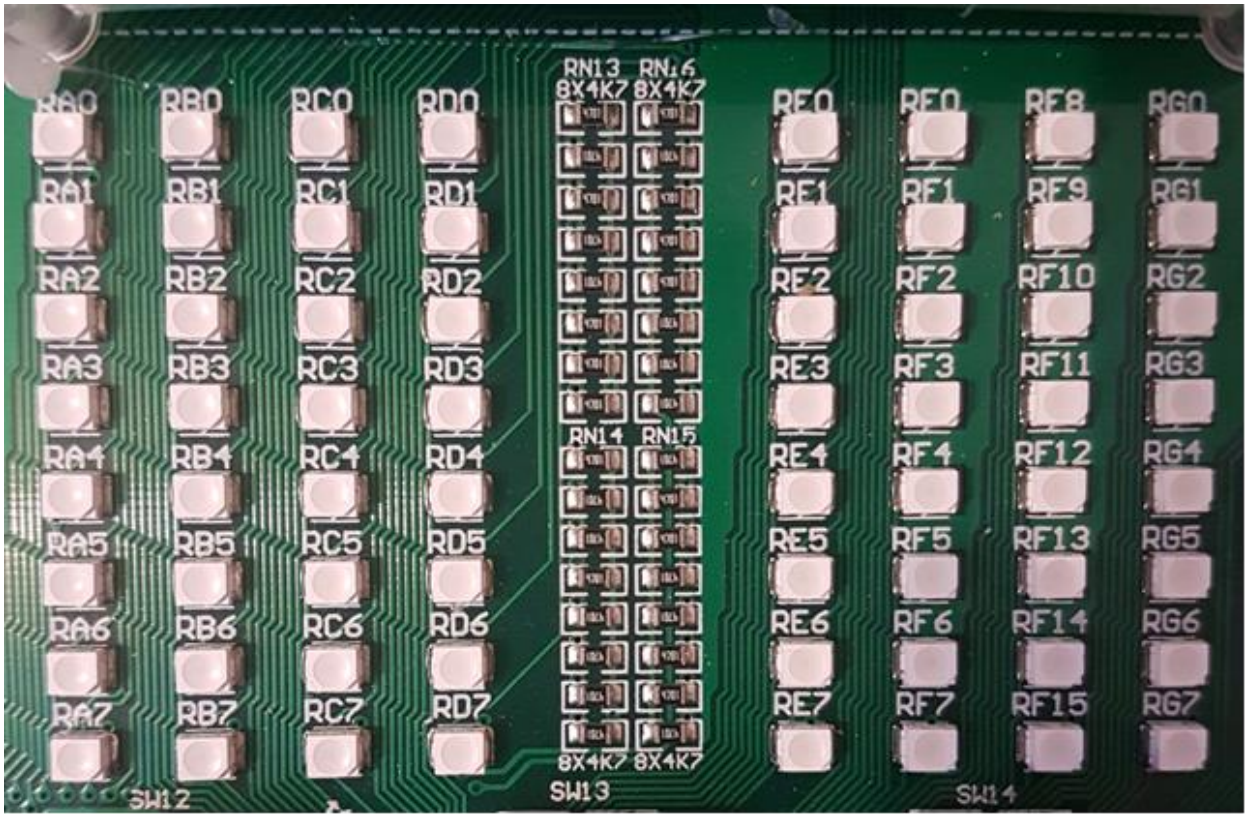
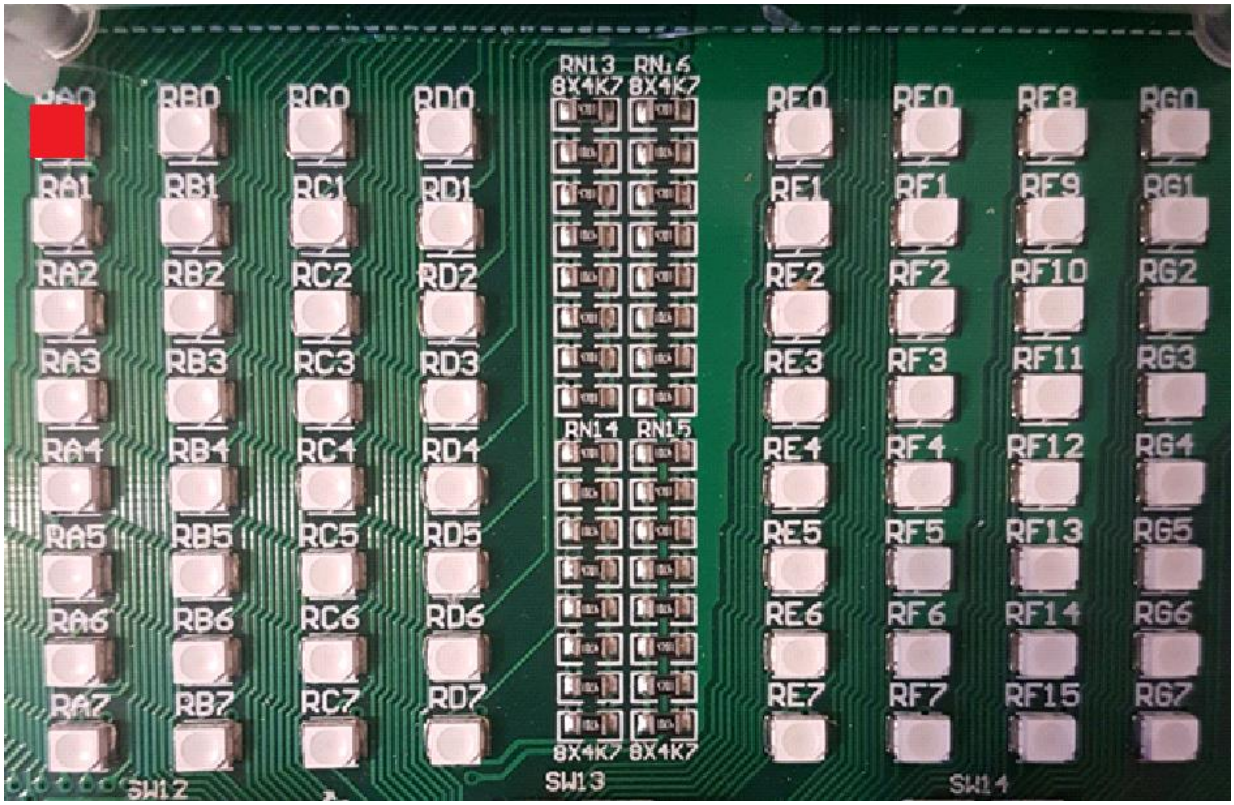
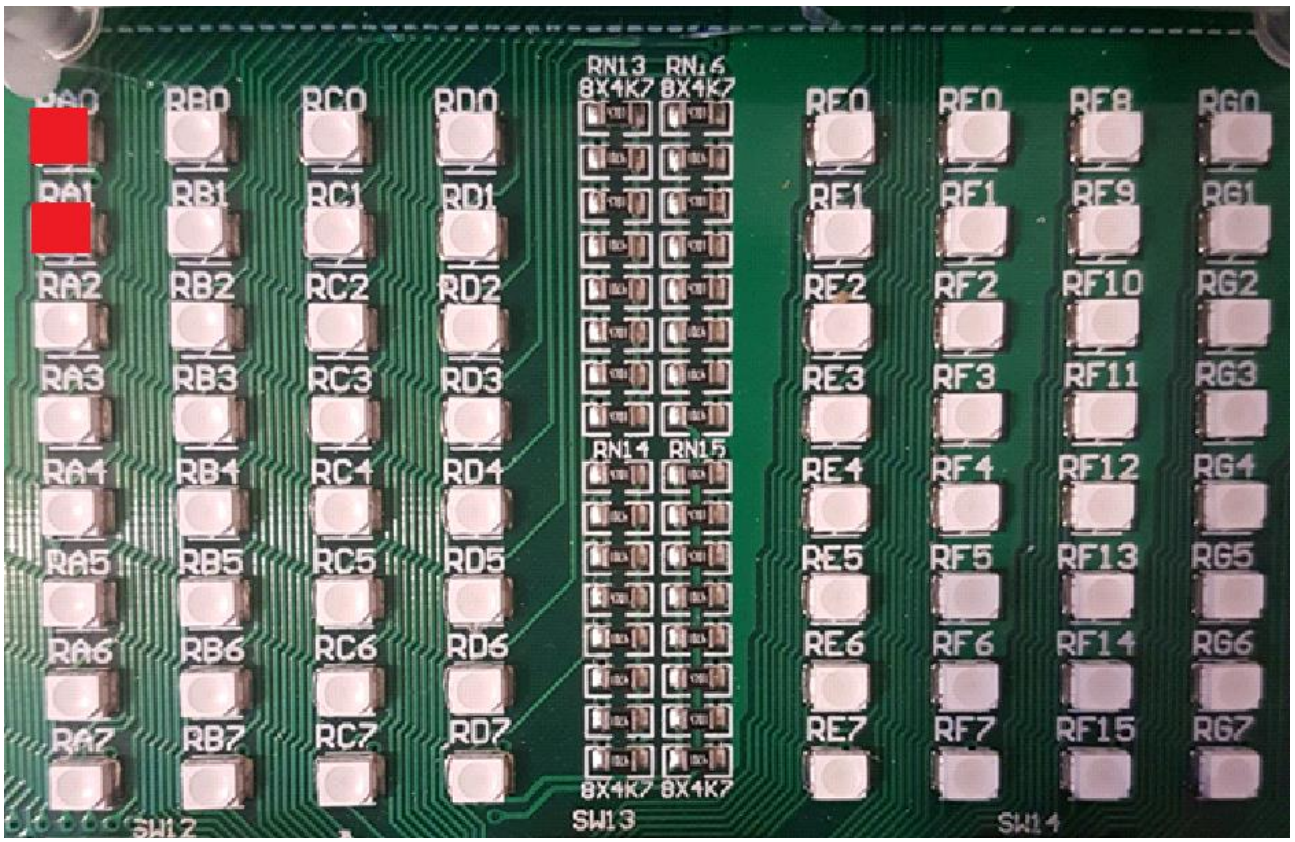
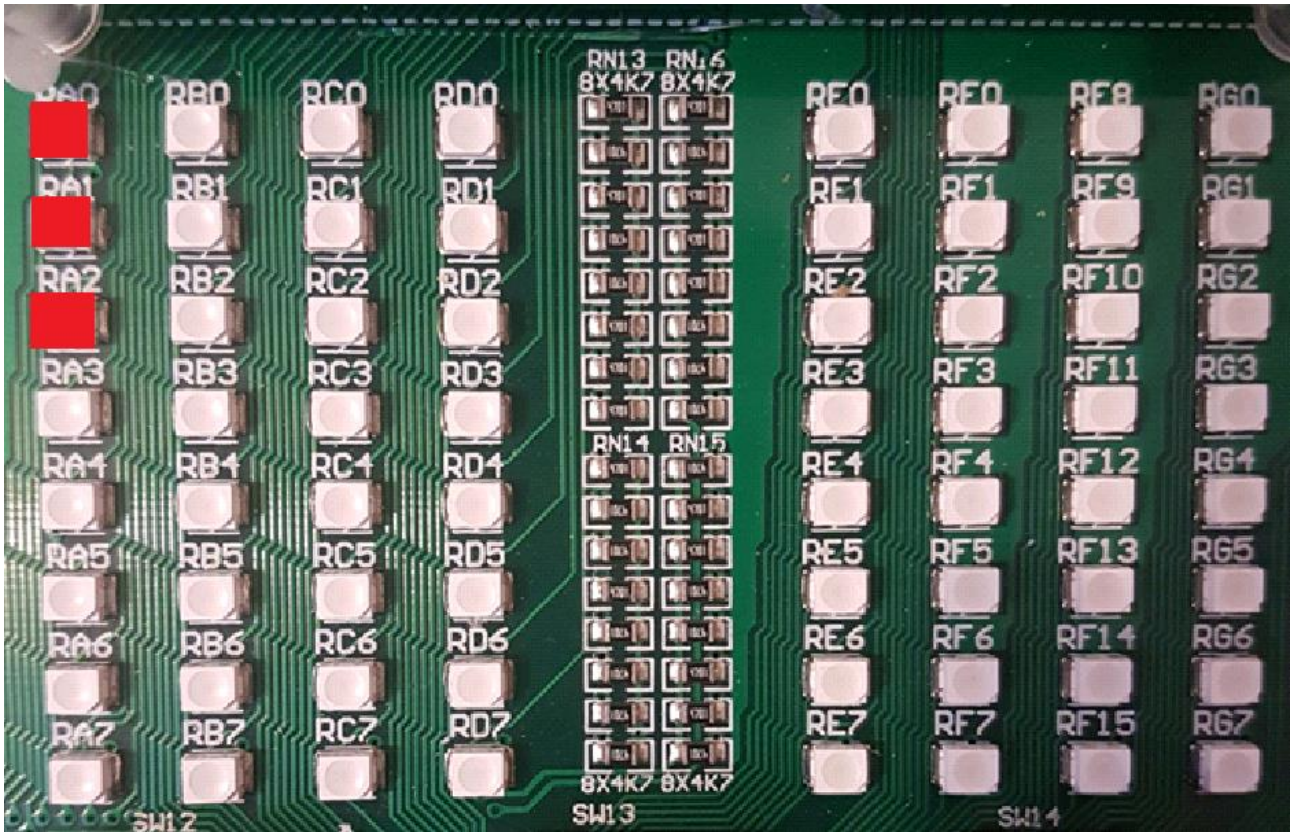
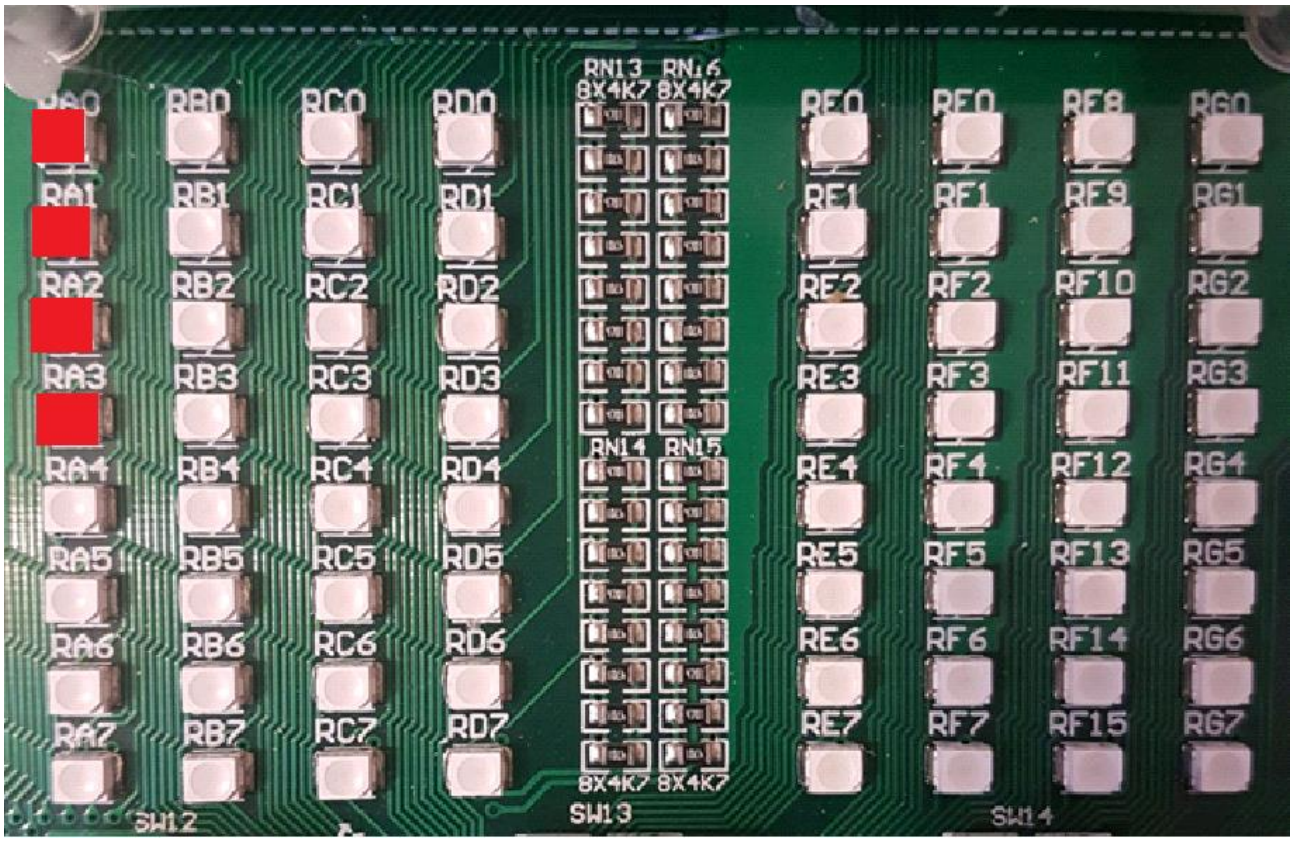


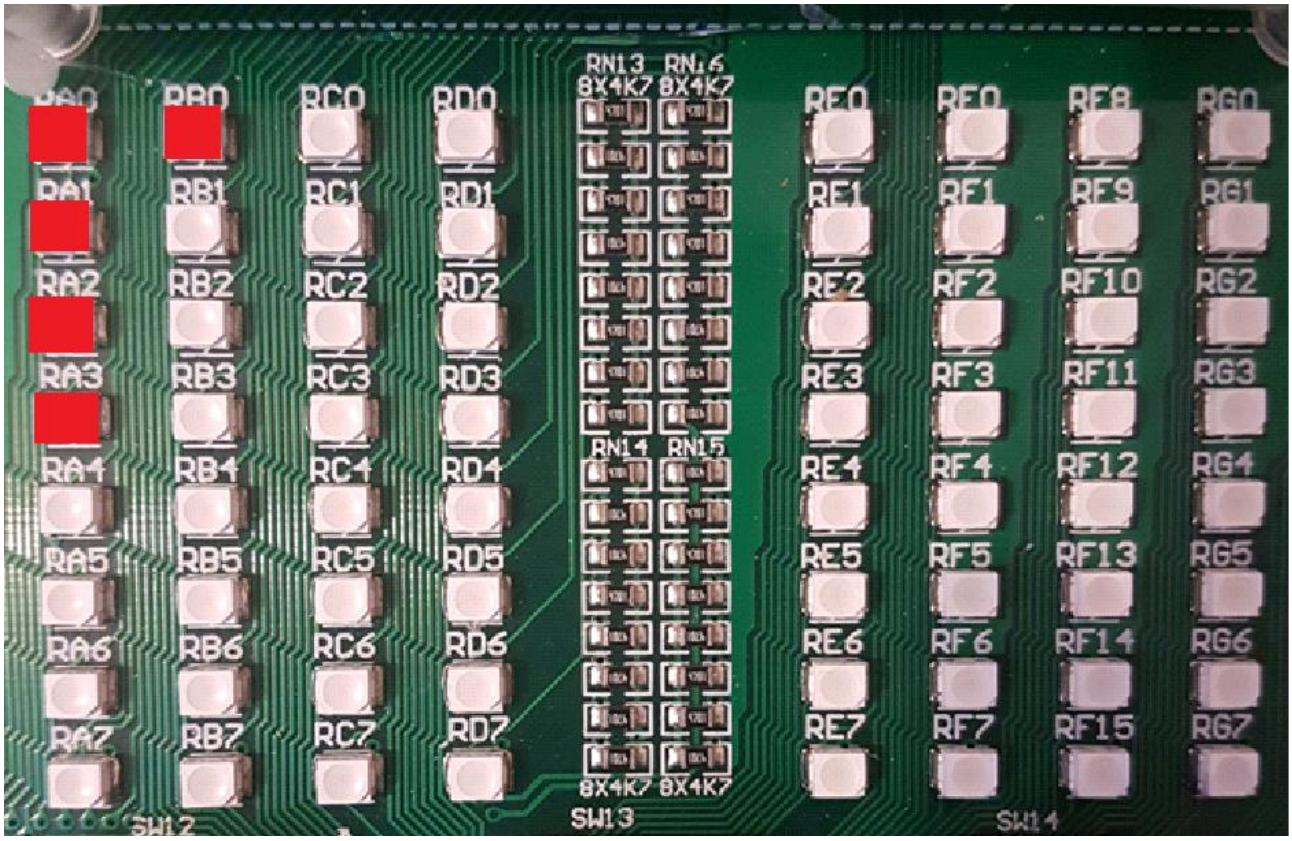
Figure 5. Turn off all LEDs for 1 second

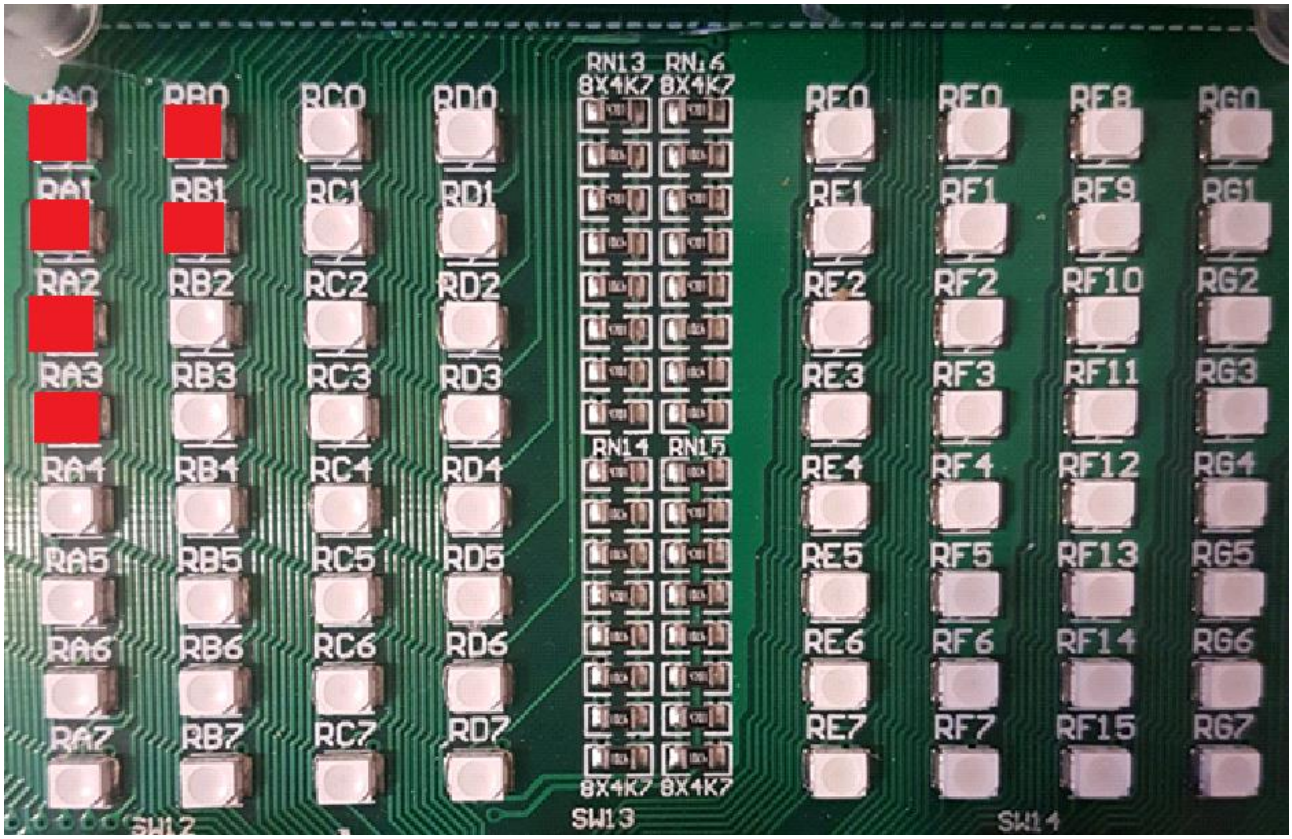

RN13 RN16
8X4K7 8X4K7
RN14 RN15
8X4K7 8X4K7
SW12
SW13
SW14

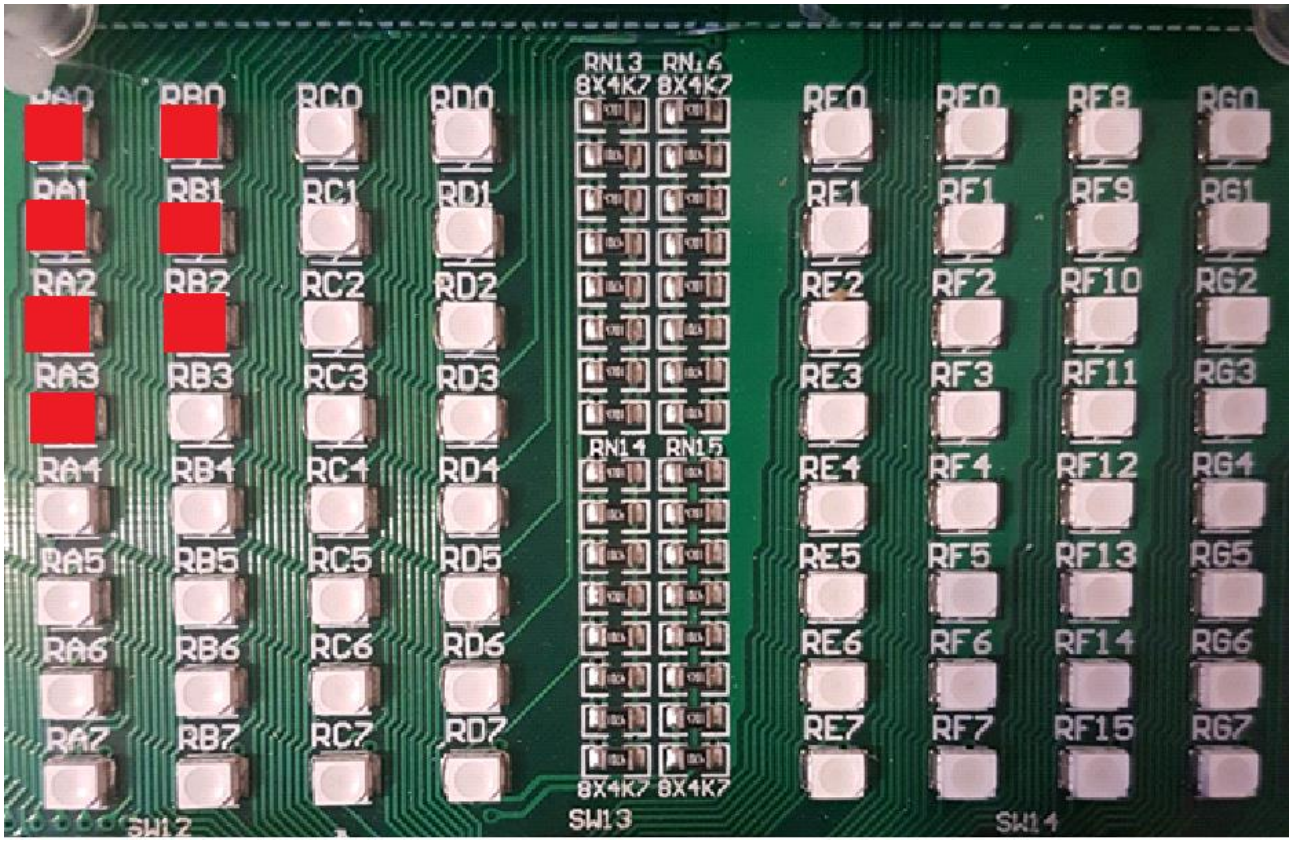

RN13 RN16
8X4K7 8X4K7
RN14 RN15
8X4K7 8X4K7
SW12
SW13
SW14

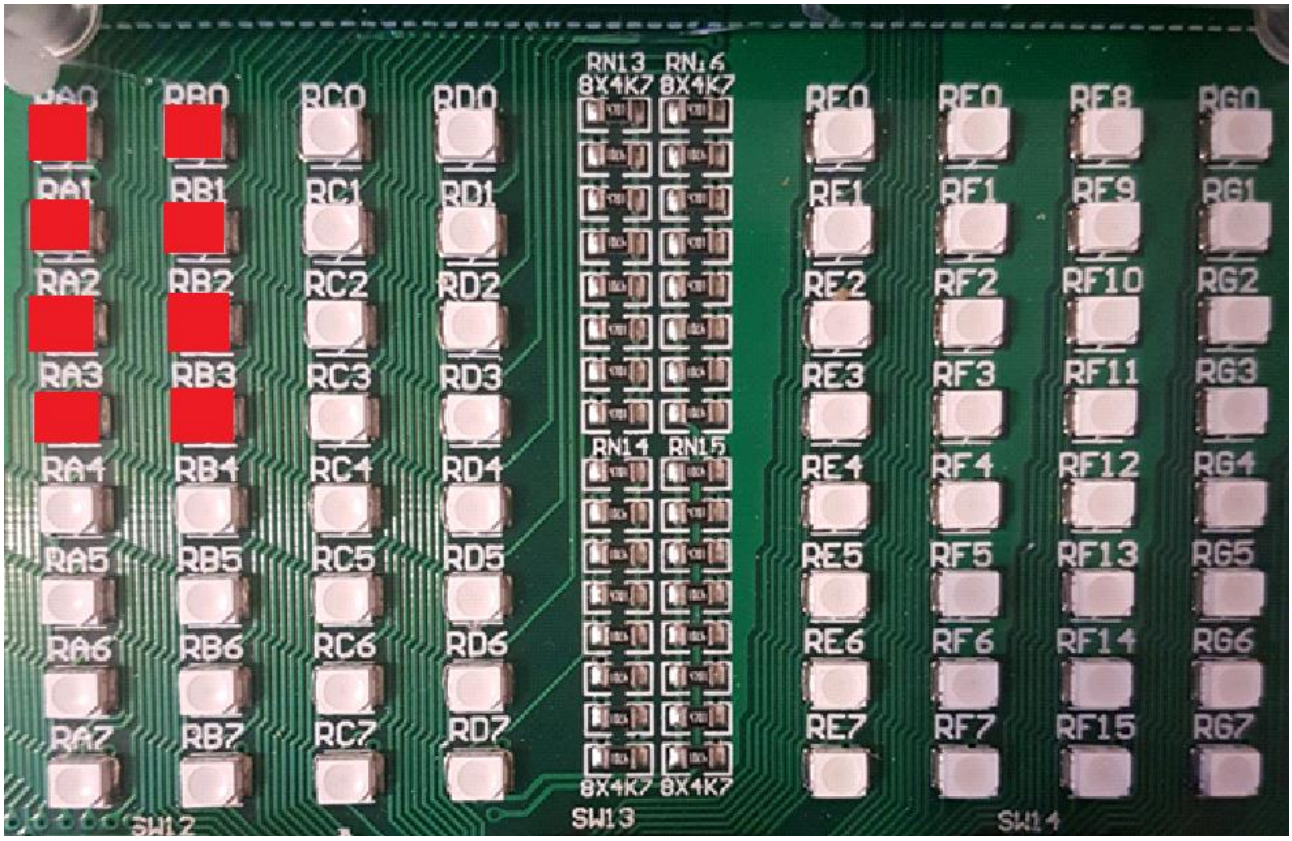

RN13 RN16
8X4K7 8X4K7
RB0 RC0 RD0 RE0 RF0 RF8 RG0
RA1 RB1 RC1 RD1 RE1 RF1 RF9 RG1
RA2 RB2 RC2 RD2 RE2 RF2 RF10 RG2
RA3 RB3 RC3 RD3 RE3 RF3 RF11 RG3
RN14 RN15
RA4 RB4 RC4 RD4 RE4 RF4 RF12 RG4
RA5 RB5 RC5 RD5 RE5 RF5 RF13 RG5
RA6 RB6 RC6 RD6 RE6 RF6 RF14 RG6
RA7 RB7 RC7 RD7 RE7 RF7 RF15 RG7
8X4K7 8X4K7
SW12
SW13
SW14


RN13 RN16
8X4K7 8X4K7
RB0 RC0 RD0 RE0 RF0 RF8 RG0
RA1 RB1 RC1 RD1 RE1 RF1 RF9 RG1
RA2 RB2 RC2 RD2 RE2 RF2 RF10 RG2
RA3 RB3 RC3 RD3 RE3 RF3 RF11 RG3
RN14 RN15
RA4 RB4 RC4 RD4 RE4 RF4 RF12 RG4
RA5 RB5 RC5 RD5 RE5 RF5 RF13 RG5
RA6 RB6 RC6 RD6 RE6 RF6 RF14 RG6
RA7 RB7 RC7 RD7 RE7 RF7 RF15 RG7
8X4K7 8X4K7
SW12
SW13
SW14


RN13 RN16
8X4K7 8X4K7
RA0 RB0 RC0 RD0 RE0 RF0 RF8 RG0
RA1 RB1 RC1 RD1 RE1 RF1 RF9 RG1
RA2 RB2 RC2 RD2 RE2 RF2 RF10 RG2
RA3 RB3 RC3 RD3 RE3 RF3 RF11 RG3
RN14 RN15
RA4 RB4 RC4 RD4 RE4 RF4 RF12 RG4
RA5 RB5 RC5 RD5 RE5 RF5 RF13 RG5
RA6 RB6 RC6 RD6 RE6 RF6 RF14 RG6
RA7 RB7 RC7 RD7 RE7 RF7 RF15 RG7
8X4K7 8X4K7
SW12
SW13
SW14


RN13 RN16
8X4K7 8X4K7
RA0 RB0 RC0 RD0 RE0 RF0 RF8 RG0
RA1 RB1 RC1 RD1 RE1 RF1 RF9 RG1
RA2 RB2 RC2 RD2 RE2 RF2 RF10 RG2
RA3 RB3 RC3 RD3 RE3 RF3 RF11 RG3
RN14 RN15
RA4 RB4 RC4 RD4 RE4 RF4 RF12 RG4
RA5 RB5 RC5 RD5 RE5 RF5 RF13 RG5
RA6 RB6 RC6 RD6 RE6 RF6 RF14 RG6
RA7 RB7 RC7 RD7 RE7 RF7 RF15 RG7
8X4K7 8X4K7
SW12
SW13
SW14


RN13 RN16
8X4K7 8X4K7
RN14 RN15
8X4K7 8X4K7
SW13
SW14
RA0 RB0 RC0 RD0 RE0 RF0 RF8 RG0
RA1 RB1 RC1 RD1 RE1 RF1 RF9 RG1
RA2 RB2 RC2 RD2 RE2 RF2 RF10 RG2
RA3 RB3 RC3 RD3 RE3 RF3 RF11 RG3
RA4 RB4 RC4 RD4 RE4 RF4 RF12 RG4
RA5 RB5 RC5 RD5 RE5 RF5 RF13 RG5
RA6 RB6 RC6 RD6 RE6 RF6 RF14 RG6
RA7 RB7 RC7 RD7 RE7 RF7 RF15 RG7


RN13 RN16
8X4K7 8X4K7
RN14 RN15
8X4K7 8X4K7
SW13
SW14
RA0 RB0 RC0 RD0 RE0 RF0 RF8 RG0
RA1 RB1 RC1 RD1 RE1 RF1 RF9 RG1
RA2 RB2 RC2 RD2 RE2 RF2 RF10 RG2
RA3 RB3 RC3 RD3 RE3 RF3 RF11 RG3
RA4 RB4 RC4 RD4 RE4 RF4 RF12 RG4
RA5 RB5 RC5 RD5 RE5 RF5 RF13 RG5
RA6 RB6 RC6 RD6 RE6 RF6 RF14 RG6
RA7 RB7 RC7 RD7 RE7 RF7 RF15 RG7

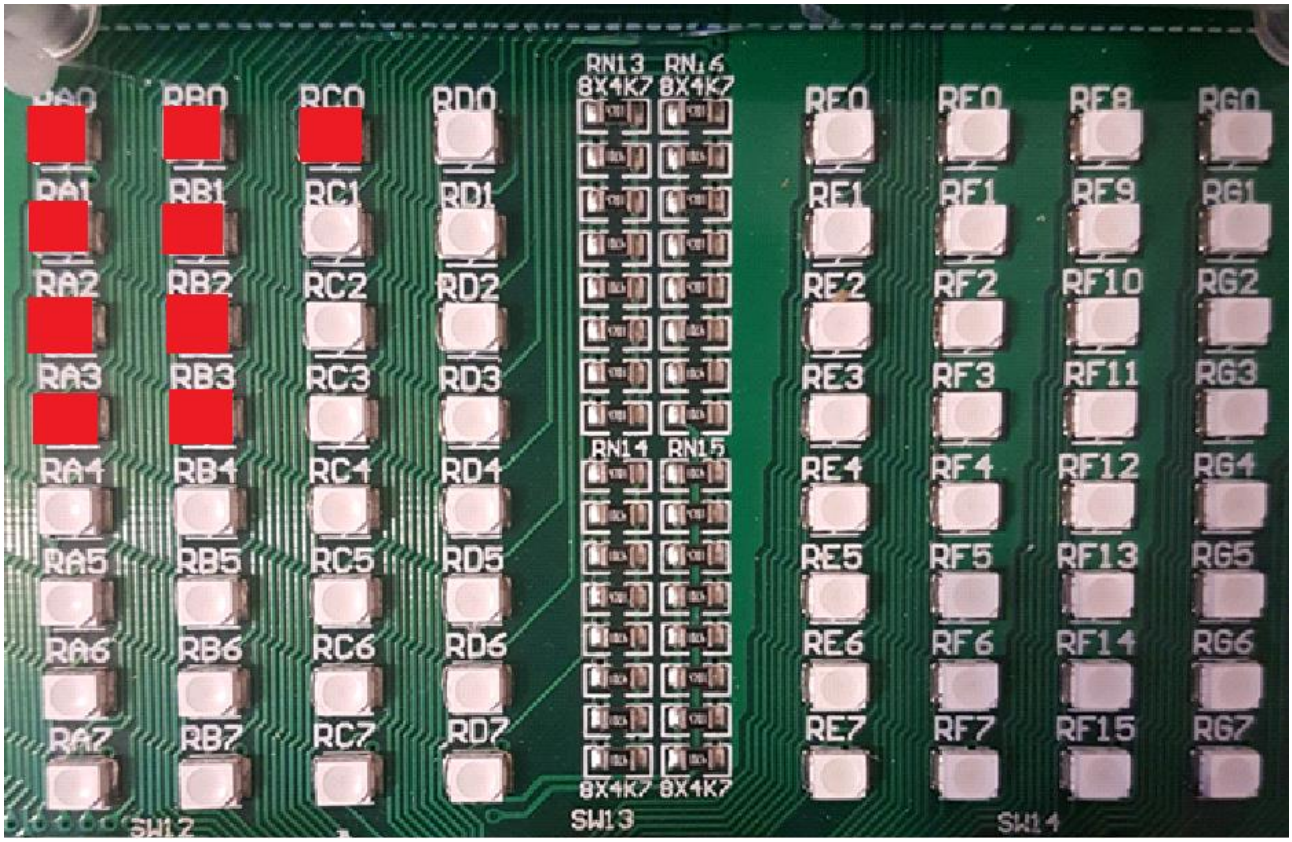


Figure 6. Samples of State1 are showed in the above pictures

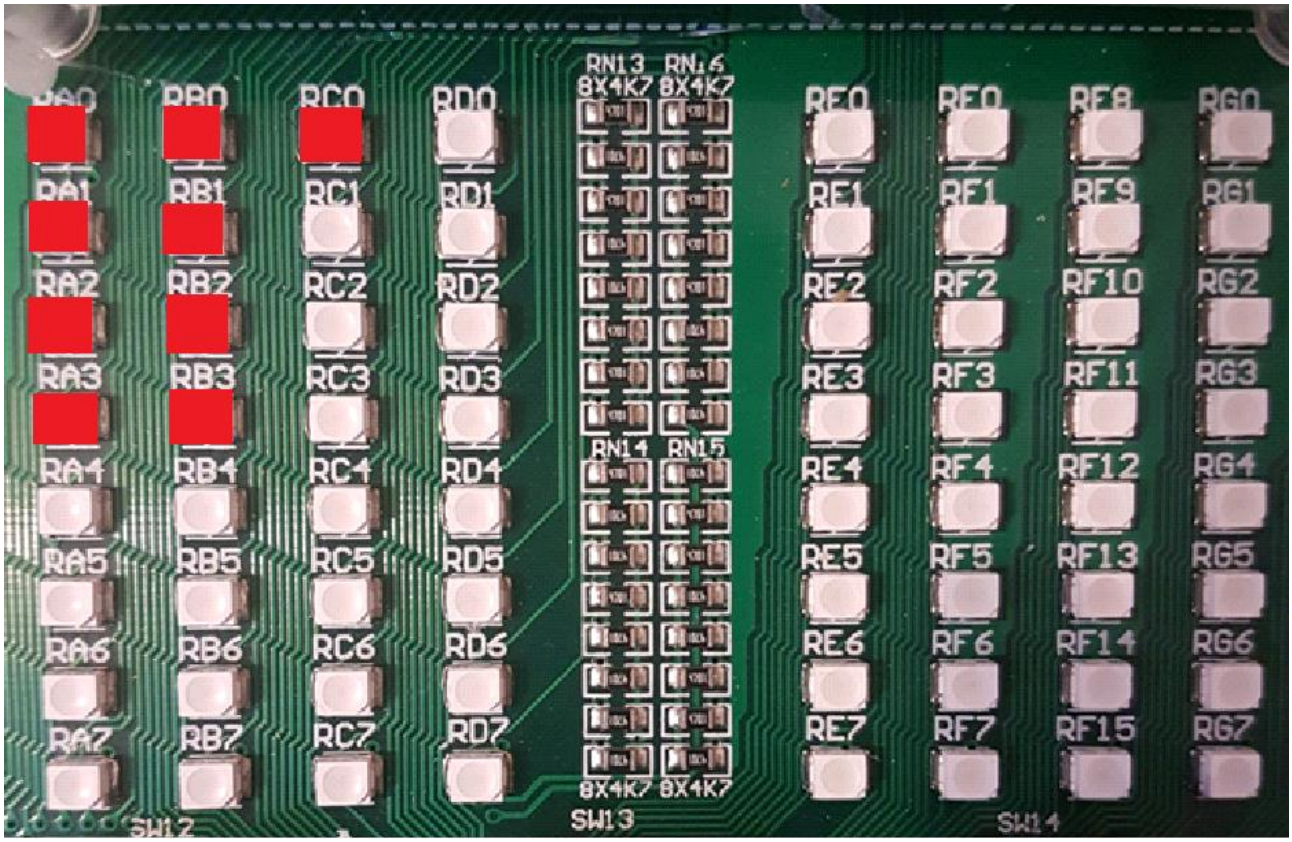
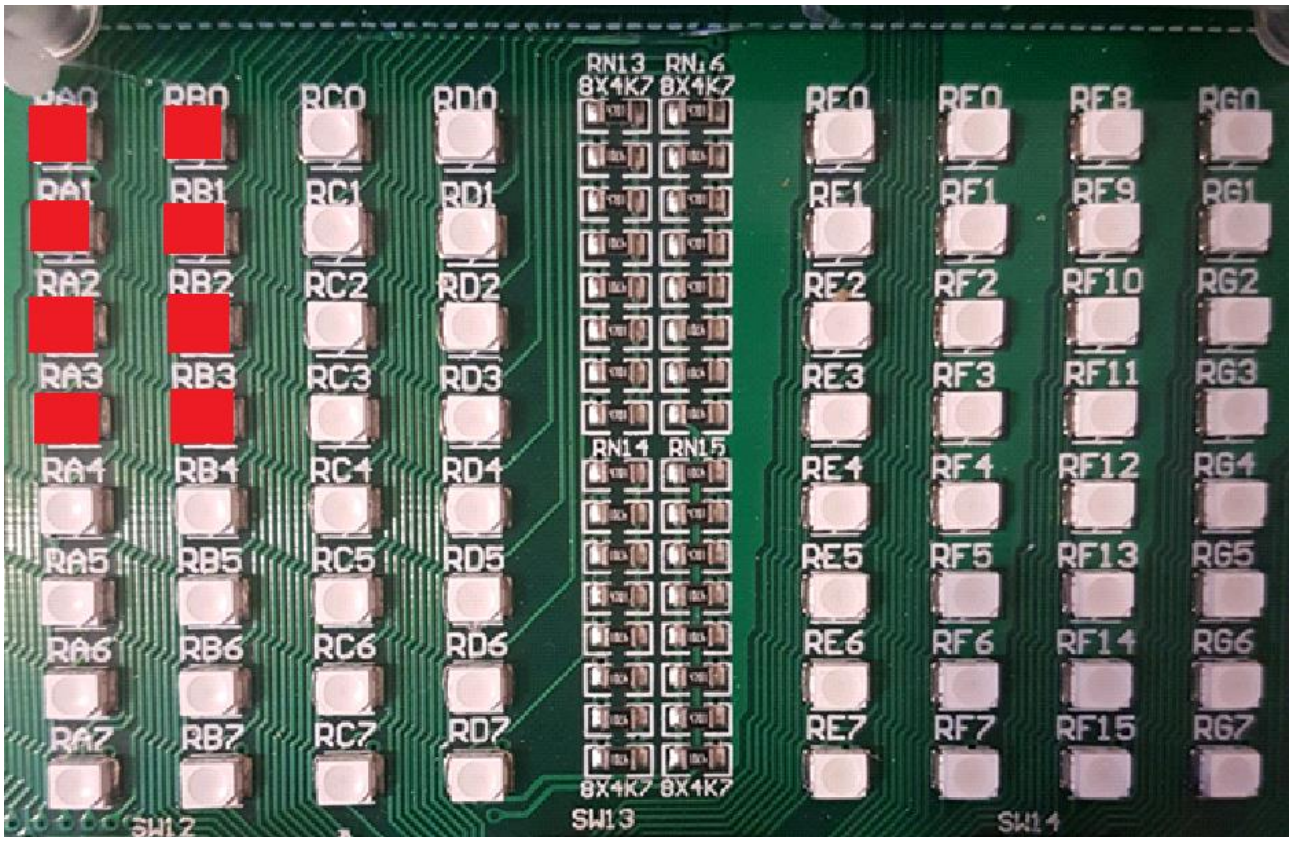


Figure 7. Sample of State2



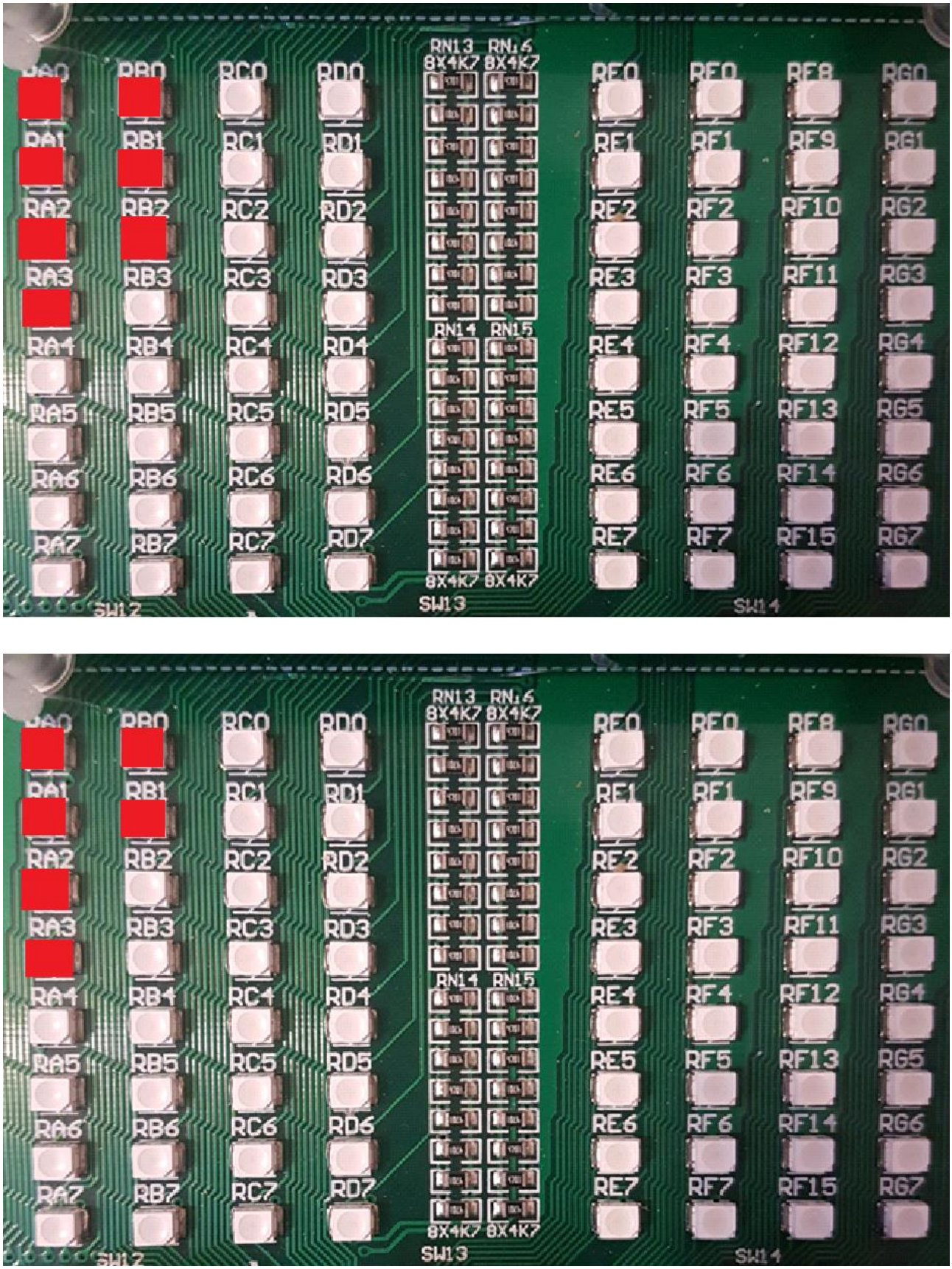


Figure 8. Samples of State3 are showed in the above pictures