

**REPORT**

**MICROPROCESSOR & MICROCONTROLLER**

**(SMJE 3183)**

**Assignment 4: Switch Blinking**

Name : Nur Azreena binti Md Azman

: Muhammad Hafizuddin bin Redzuan

: Nur Aida Hamizah binti Kamarudin

Course/Section : 3 SMJE/02

Group : 5

Lecturer : Dr Azura & Dr Husna

**INTRODUCTION:**

For this project, we are required to do a task which is switch blinking. This program needs us to do a program where 8 LEDs are switching from mode 1 to mode 2. We programmed mode 1 to do a full blinking and mode 2 we programmed to do a half blinking. These 2 modes will switch automatically.

**OBJECTIVE:**

* To be able to write programs with the task given.
* To be able to run the programs written based on the task given on the microcontroller board kit.

**TASK:**

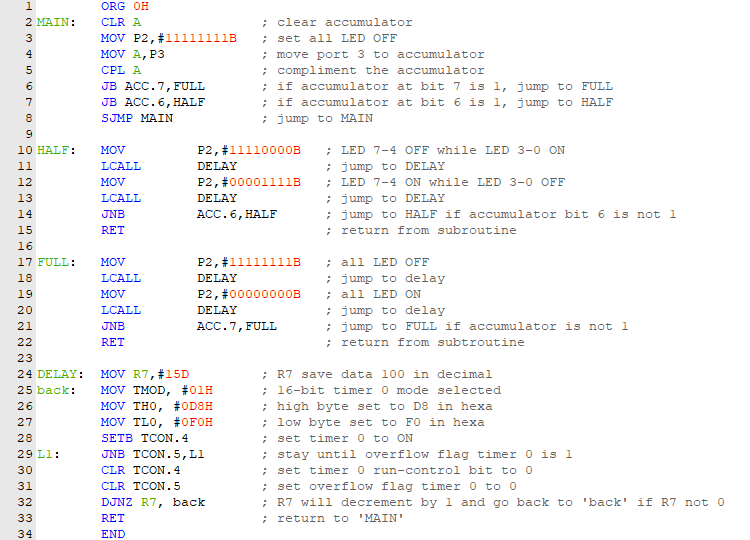
Students are required to write a programming for Switch Blinking and run the program and the output will show on the microcontroller board kit.

**EQUIPMENTS AND COMPONENTS:**

* Microcontroller circuit board.
* Keil Embedded Programs.

**RESULT AND SIMULATION:**

Switching control full and half blinking LED



1. LED is control by port 2 from LED 0 to LED 7 while Switch 1 to Switch 6 is control by port 3.7 to port 3.2 respectively.
2. Port 2 is set 11111111B for let all LED in condition OFF by default.
3. Port 3 is move to accumulator in order to complement the value of Port 3 since it is active low input
4. The instruction will jump to the FULL or HALF label depend on accumulator bit 7 and 6 is set respectively.
5. If the accumulator bit 7 is set, label FULL instruction will be executed. For FULL instruction:

* Port 2 is set 11111111B, since the LED is active-LOW so all the LED will OFF
* DELAY is called so it will jump to DELAY to be executed to next instruction
* R7 data is set to 15 in decimal
* Timer mode will set to 01 in hexa to select 16-bit timer 0
* For crystal oscillator, there need to make timer count up to 10 000 since 1 machine cycles is 1µs. So, 65 536 (full timer count for crystal oscillator) must minus 10 000, 65 536 – 10 000 = 55 536 = DBF0H
* TH0 is for high byte set so DB will be set while TH1 is for low byte set so F0 will be set
* TCON.4 is for Timer 0 run-control bit so it will be set 1 to timer start counting
* The instruction for timer count will keep remain until overflow flag is 1. In other words, Timer 0 finish counting.
* TCON.4 and TCON.5 will set back to 0 to be ready to execute next instruction
* R7 will decrement by 1 so, R7 data now is 14 and instruction will execute ‘back’ until R7 decrease until 0.
* After R7 become 0 the instruction will return from subroutine where DELAY is called and will execute Line 19. This will give all LED OFF for 15 x 10ms = 0.15s
* Then, the same cycle will be execute but for Port 2 00000000B. So, all LED will ON for another 0.15s. This will let the all LED blinking
* After 1 cycle full blinking completed, the instruction will check if Switch 1 is still active or not (press or release) and it will execute back full blinking instruction if Switch 1 is still active.
* If Switch 1 is not active (not press) the subroutine will return where FULL label is called

1. If the accumulator bit 7 is not set while accumulator bit 6 is set, label HALF instruction will be executed. For HALF instruction:

* Port 2 is set 11110000B since the LED is active-LOW so LED 7-4 OFF while LED 3-1 ON
* DELAY is called so it will jump to DELAY to be executed to next instruction
* R7 data is set to 15 in decimal
* Timer mode will set to 01 in hexa to select 16-bit timer 0
* For crystal oscillator, there need to make timer count up to 10 000 since 1 machine cycles is 1µs. So, 65 536 (full timer count for crystal oscillator) must minus 10 000, 65 536 – 10 000 = 55 536 = DBF0H
* TH0 is for high byte set so DB will be set while TH1 is for low byte set so F0 will be set
* TCON.4 is for Timer 0 run-control bit so it will be set 1 to timer start counting
* The instruction for timer count will keep remain until overflow flag is 1. In other words, Timer 0 finish counting.
* TCON.4 and TCON.5 will set back to 0 to be ready to execute next instruction
* R7 will decrement by 1 so, R7 data now is 14 and instruction will execute ‘back’ until R7 decrease until 0.
* After R7 become 0 the instruction will return from subroutine where DELAY is called and will execute Line 12. This will give half LED OFF for 15 x 10ms = 0.15s
* Then, the same cycle will be execute but for Port 2 00001111B. So, LED 7-6 will ON while LED 4-0 will OFF for another 0.15s. This will let the half LED blinking
* After 1 cycle half blinking completed, the instruction will check if Switch 2 is still active or not (press or release) and it will execute back full blinking instruction if Switch 2 is still active.
* If Switch 2 is not active (not press) the subroutine will return where HALF label is called

1. If any of the switch not active (press), the instruction will jump back to MAIN and this will let LED in OFF condition as default

**CONCLUSION:**

Conclusions, we programmed our task by using timer function and the output of the program that we achieved is by a long pressing on the switch to switch it instead of the switching happens automatically.