Car-park routing system

I have implemented this scenario using a single FSM built by switch statements with 46 states in total.

- For Communication, UART is sourced from ACLK (32768Hz) with baud rate generator 9600 where the Prefix is CPRS_ and Suffix is \CR\LF and defined all necessary configurations. (RES, DIR1, DIR2, OPN, CLS, CNT, SPD, MAX123, CLKXX:XX, CLSXX:XX, OPNXX:XX)
- Ring buffer of size 32 is used to store and parse the characters received by the device one after the other in a sequence manner by incrementing Head position (Push operation) and Tail position (Pop operation) in UART interrupt and main function respectively.

```
Initial parameters of the parking lot:
vehicle_count = 0;  // Counting register for number of vehicles.
max_limit = 255;    // Maximum vehicles(Default).
avg_speed = 0;      // Average speed of all vehicles inside the garage(KMPH)
direction = 0;      // Direction: Forward 0(default), Reverse 1.
PL_Status = PL_OPEN;// Parking Lot Status: Open(Default).
```

- For an incoming vehicle(button1 event prior to button2), I check if the PL_Status == PL_OPEN in the port interrupt and if so after successful debouncing and entering into Timer TA0 interrupt, I check if the parking lot is full or is vacant and only if vehicle_count < max_limit I increment vehicle_count and add the speed of that vehicle to the avg_speed and also blink the GREEN LED for 200ms using TA1 Timer in compare mode (after the speed calculation in capture mode).
- Similar operation for an outgoing vehicle(button2 event prior to button1).
- Speed Measurement example: On a successful button1 press event, I initialise and start the Timer TA1 and capture its value on the immediate successful button2 press event and then use this Time to calculate the speed of the vehicle (Assumed distance between the sensors to be 1 metre for practical speed values in KMPH as button action(by humans) cannot match the true speed of the car).

```
    // speed = distance / time; (Minimum speed = 2 KMPH)
    // distance = 1 metre = 0.001 km (For practical speed values)
    // time = (TA1CCR/10,000)sec .... (from ACLCK 10KHz - 1 sec)
    // time = (TA1CCR/(10,000*3600))hr
```

- When MAX123 (123 → any number<=255) is sent, the max vehicle limit is set to that value and vehicle_count cannot increment more than that value.
- On switching the direction to DIR2, incoming and outgoing vehicles are reversed so that counter and LED blinking functions in the opposite manner.
- When RES is sent, all the parameters of the parking lot will be reset to the default values and existing automatic close and open alarms will be removed.
- Real Time Clock RTC is sourced from the Low Frequency oscillator LFXT at 32768 Hz and the initial time is set as 08:00 by default. By sending CLSXX:XX, user can initialise the RTC again with any new time as mentioned. By using CLSXX:XX and OPNXX:XX, user can set automatic opening and closing times of the parking lot (which when executed blink both the LEDs for 400ms as a acknowledgement for CLOSE/OPEN.)
- As UART, Timers and RTC are sourced from ACLCK only, most of the time device can run in the LPM3 mode where only ACLK is active and only when CPU is required (for push operations and parse calling), device needs to run in Active mode. This can be achieved by switching to LPM3 when Head position equals Tail position and switching back to Active in UART interrupt when any character is received(before popping).

- Kaushik Goud Chandapet (1536199)