Moving Car

By Omar Ashraf Taha
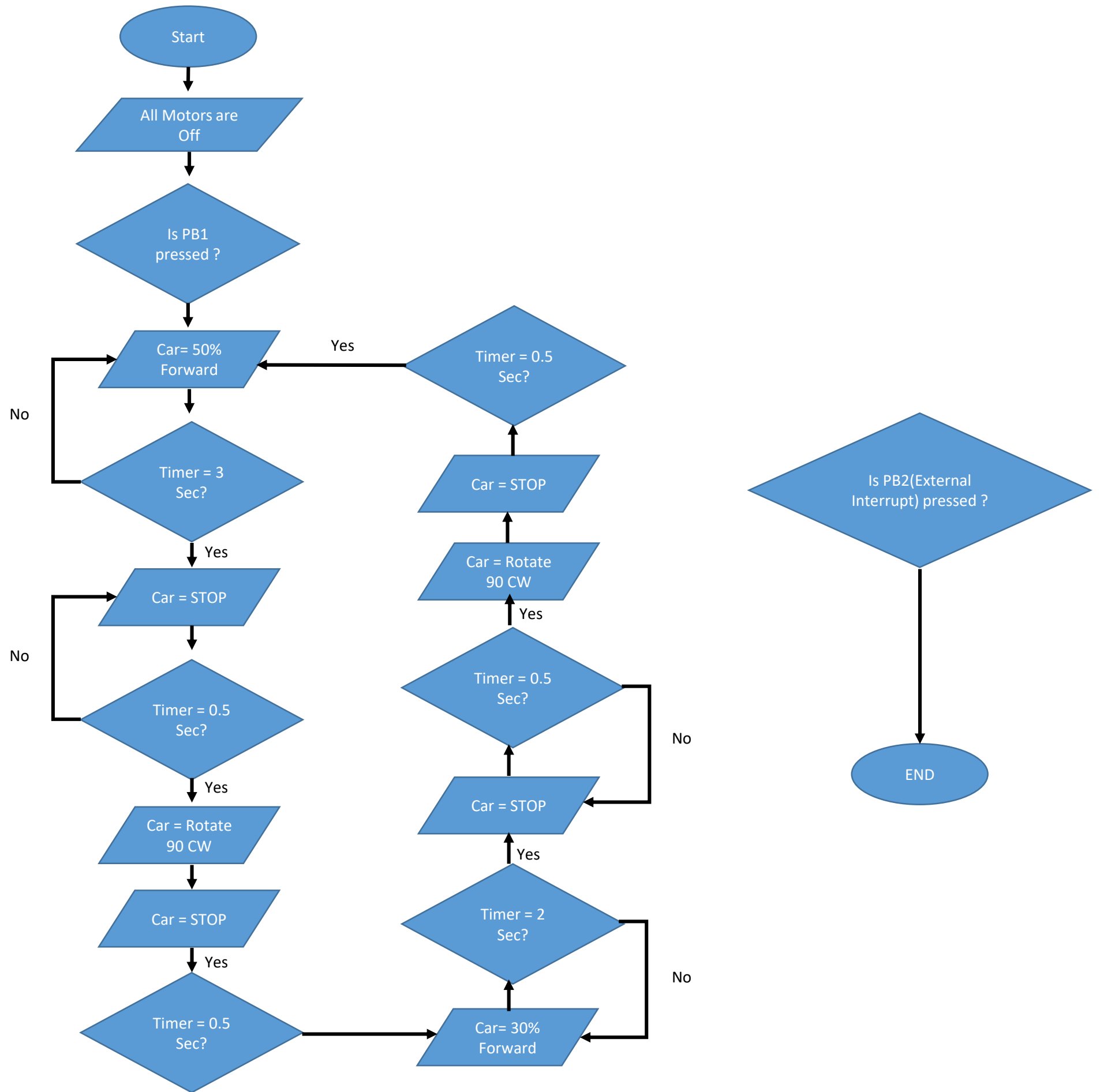
Moving Car

Description :

1. **Car Components**:
    1. Four motors (M1, M2, M3, M4)
    2. One button to start (PB1)
    3. One button for stop (PB2)
    4. Four LEDs (LED1, LED2, LED3, LED4)

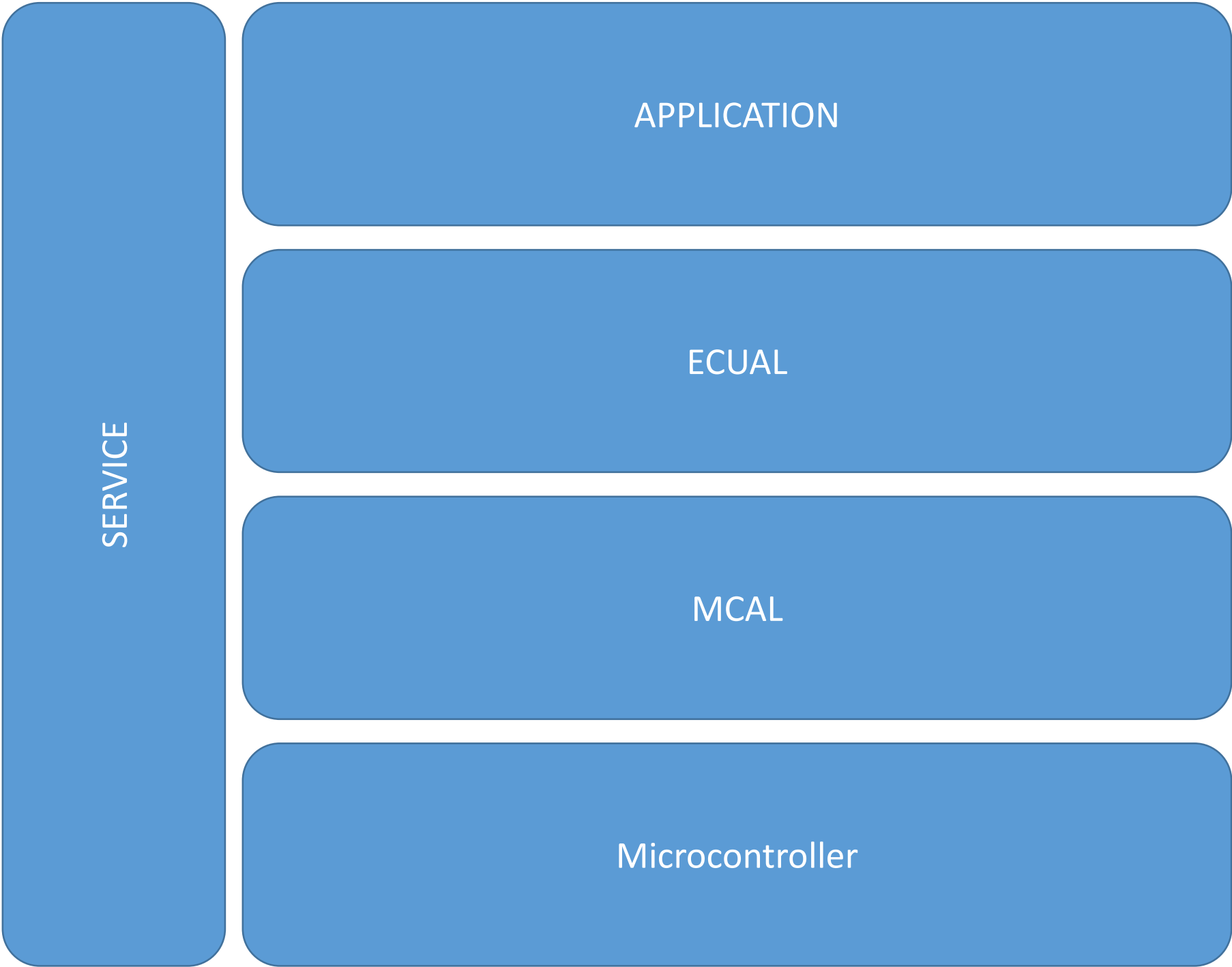2. **System Requirements**:
    1. The car starts initially from 0 speed
    2. When PB1 is pressed, the car will move forward after 1 second
    3. The car will move forward to create the longest side of the rectangle for 3 seconds with 50% of its maximum speed
    4. After finishing the first longest side the car will stop for 0.5 seconds, rotate 90 degrees to the right, and stop for 0.5 second
    5. The car will move to create the short side of the rectangle at 30% of its speed for 2 seconds
    6. After finishing the shortest side the car will stop for 0.5 seconds, rotate 90 degrees to the right, and stop for 0.5 second
    7. Steps 3 to 6 will be repeated infinitely until you press the stop button (PB2)
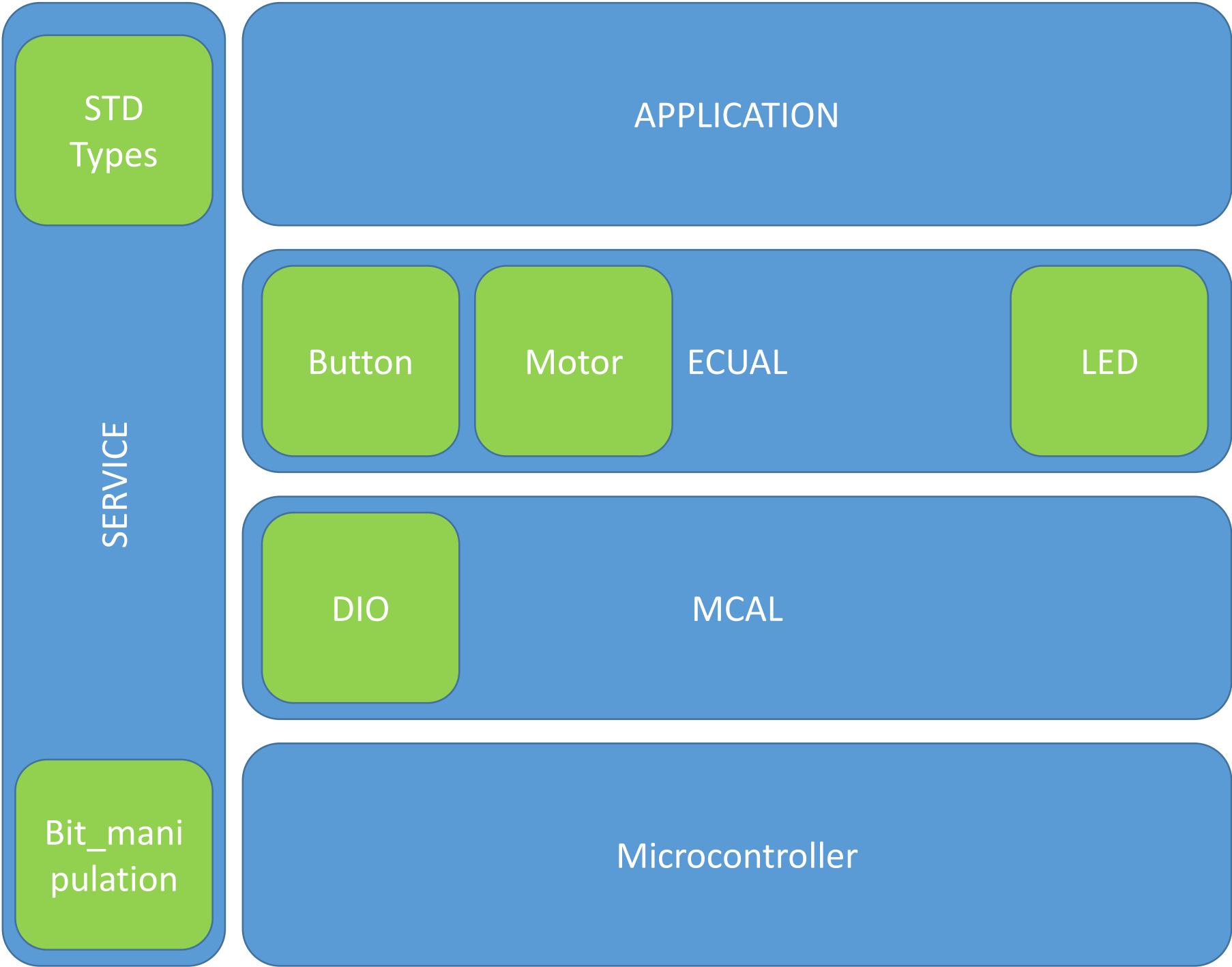   8. PB2 acts as a sudden break, and it has the highest priority

Project flowchart:

```
                              Start
                                │
                                ▼
                        ┌─────────────────┐
                        │ All Motors are  │
                        │      Off        │
                        └─────────────────┘
                                │
                                ▼
                          ◇ Is PB1     ◇
                          ◇ pressed ?  ◇
                                │
                                ▼
      ┌────────────►  Car= 50%            ◄──── Yes ──── ◇ Timer = 0.5 ◇
      │               Forward                            ◇   Sec?      ◇
      │                   │                                    ▲
      No                  ▼                                    │
      │            ◇ Timer = 3 ◇                         Car = STOP
      └────────────◇   Sec?    ◇                               ▲
                         │                                     │
                        Yes                             Car = Rotate
                         ▼                                90 CW
      ┌────────────► Car = STOP                               ▲
      │                  │                                   Yes
      No                 ▼                                    │
      │          ◇ Timer = 0.5 ◇                      ◇ Timer = 0.5 ◇──── No
      └──────────◇    Sec?     ◇                      ◇    Sec?     ◇     │
                        │                                   ▲            │
                       Yes                                  │            │
                        ▼                            Car = STOP ◄────────┘
                   Car = Rotate                             ▲
                     90 CW                                 Yes
                        │                                   │
                        ▼                            ◇ Timer = 2 ◇──── No
                   Car = STOP                        ◇   Sec?    ◇     │
                        │                                  ▲           │
                       Yes                                 │           │
                        ▼                                  │           │
              ◇ Timer = 0.5 ◇ ──────────►  Car= 30% ◄──────────────────┘
              ◇    Sec?     ◇              Forward
```

◇ Is PB2(External Interrupt) pressed ? ◇
                │
                ▼
             END

Layered Architecture:

APPLICATION

ECUAL

SERVICE

MCAL

Microcontroller

Modules/Drivers:

APIs:

Motor Driver:

- err_state MOTOR_init(uint8_t pinNumber, uint8_t portNumber);

- err_state MOTOR_on(uint8_t pinNumber, uint8_t portNumber);

- err_state MOTOR_off(uint8_t pinNumber, uint8_t portNumber);

- err_state MOTOR_control(uint8_t pinNumber, uint8_t portNumber,uint8_t speedPercentage);

Button Driver:

- err_state BUTTON_init(uint8_t pinNumber, uint8_t portNumber);

- err_state BUTTON_read(uint8_t pinNumber, uint8_t portNumber, uint8_t *value);

LED Driver:

- err_state LED_init(uint8_t ledPin, uint8_t ledPort);

- err_state LED_on(uint8_t ledPin, uint8_t ledPort);

- err_state LED_off(uint8_t ledPin, uint8_t ledPort);

- err_state LED_toggle(uint8_t ledPin, uint8_t ledPort);

DIO Driver:

- err_state DIO_init(uint8_t pinNumber, uint8_t portNumber, pin_dir direction);

- err_state DIO_write(uint8_t pinNumber, uint8_t portNumber, pin_state value);

- err_state DIO_toggle(uint8_t pinNumber, uint8_t portNumber);

- err_state DIO_read(uint8_t pinNumber, uint8_t portNumber, uint8_t *value);