at24cxx\_read\_one\_byte()函数：

1. //read one byte at the specified address of AT24CXX
2. //read\_addr: the specified address
3. //return: the byte
4. uint8\_t at24cxx\_read\_one\_byte(uint16\_t read\_addr)
5. {
6. uint8\_t temp = 0;
7. i2c\_start();
8. **if**(EE\_TYPE > AT24C16)
9. {
10. i2c\_send\_byte(0xa0);//send write command
11. i2c\_wait\_ack();
12. i2c\_send\_byte(read\_addr >> 8);//send high address
13. }
14. **else**
15. {  //send address 0xa0, write
16. i2c\_send\_byte(0xa0 + ((read\_addr/256) << 1));
17. }
18. i2c\_wait\_ack();
19. i2c\_send\_byte(read\_addr % 256);//send low address
20. i2c\_wait\_ack();
21. i2c\_start();
22. i2c\_send\_byte(0xa1);//read
23. i2c\_wait\_ack();
24. temp = i2c\_read\_byte(0);
25. i2c\_stop();//generate stop signal
26. **return** temp;
27. }

该函数是根据at24c02的random read操作时序（page22）写的。由于at24c02的内部EEPROM地址值为0~255，故i2c\_send\_byte(0xa0 + ((read\_addr/256) << 1))其实就是i2c\_send\_byte(0xa0)。



at24cxx\_write\_one\_byte()函数：

1. //write one byte at the specified address of AT24CXX
2. **void** at24cxx\_write\_one\_byte(uint16\_t write\_addr, uint8\_t data\_to\_write)
3. {
4. i2c\_start();
5. **if**(EE\_TYPE > AT24C16)
6. {
7. i2c\_send\_byte(0xa0);
8. i2c\_wait\_ack();
9. i2c\_send\_byte(write\_addr >> 8);
10. }
11. **else**
12. {
13. i2c\_send\_byte(0xa0 + ((write\_addr/256) << 1));
14. }
15. i2c\_wait\_ack();
16. i2c\_send\_byte(write\_addr % 256);
17. i2c\_wait\_ack();
18. i2c\_send\_byte(data\_to\_write);
19. i2c\_wait\_ack();
20. i2c\_stop();
21. delay\_ms(10);
22. }

