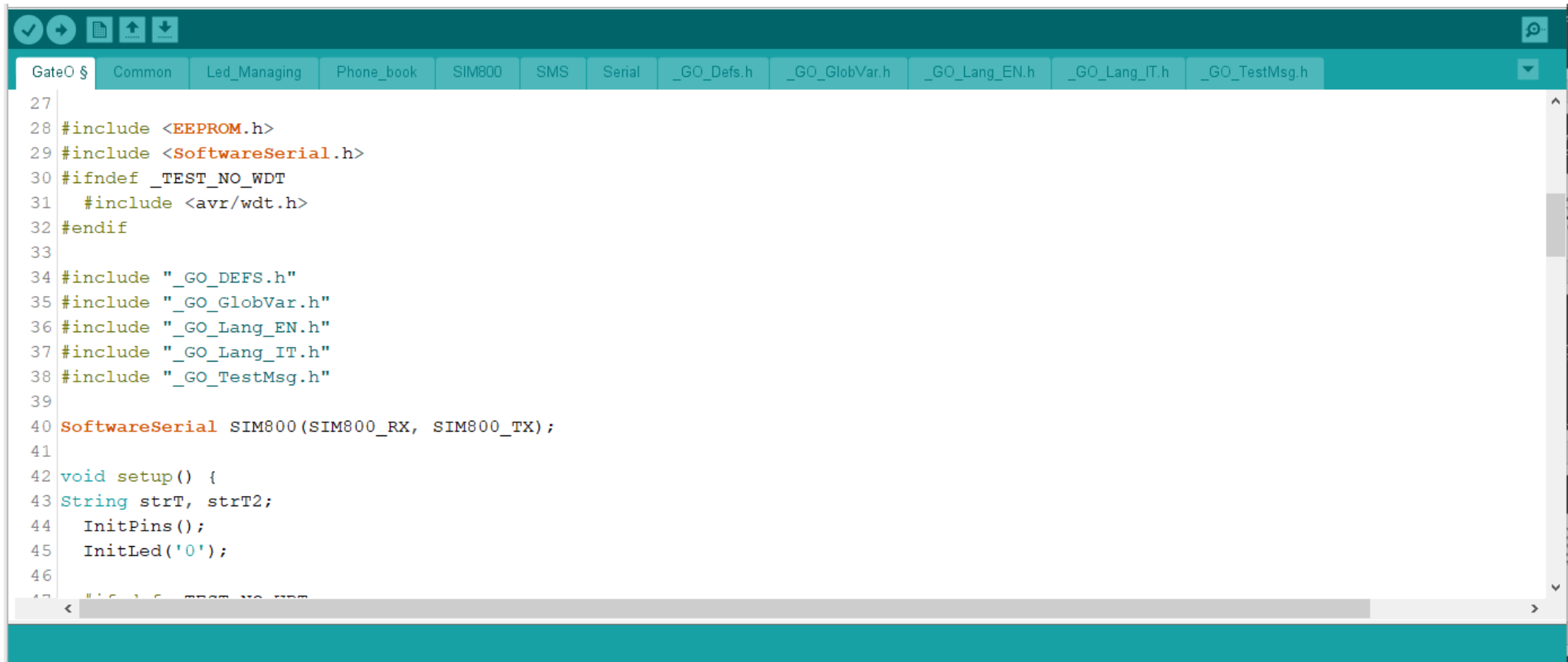


Sketch organization



The screenshot shows the Arduino IDE interface. The top toolbar contains icons for saving, running, and other functions. Below the toolbar is a tab bar with the following tabs: GateO \$, Common, Led_Managing, Phone_book, SIM800, SMS, Serial, _GO_Defs.h, _GO_GlobVar.h, _GO_Lang_EN.h, _GO_Lang_IT.h, and _GO_TestMsg.h. The main editor area displays the following code:

```
27
28 #include <EEPROM.h>
29 #include <SoftwareSerial.h>
30 #ifndef _TEST_NO_WDT
31   #include <avr/wdt.h>
32 #endif
33
34 #include "_GO_DEFS.h"
35 #include "_GO_GlobVar.h"
36 #include "_GO_Lang_EN.h"
37 #include "_GO_Lang_IT.h"
38 #include "_GO_TestMsg.h"
39
40 SoftwareSerial SIM800(SIM800_RX, SIM800_TX);
41
42 void setup() {
43   String strT, strT2;
44   InitPins();
45   InitLed('0');
46
47   #ifndef _TEST_NO_WDT
```

Sketch organization

1) Main Sketch

2) #Define

3) Global variables

5) Multi-language msgs

5) Multi-language msgs

4) Debugging macros

7) Libraries

6) .h files must be declared

```
27
28 #include <EEPROM.h>
29 #include <SoftwareSerial.h>
30 #ifndef _TEST_NO_WDT
31   #include <avr/wdt.h>
32 #endif
33
34 #include "_GO_DEFS.h"
35 #include "_GO_GlobVar.h"
36 #include "_GO_Lang_EN.h"
37 #include "_GO_Lang_IT.h"
38 #include "_GO_TestMsg.h"
39
40 SoftwareSerial SIM800(SIM800_RX, SIM800_TX);
41
42 void setup() {
43   String strT, strT2;
44   InitPins();
45   InitLed('0');
46
47   < "1) Main Sketch" >
```

Sketch organization: loop()

1) Watchdog timer reset

2) USB polling/writing

3) Phone polling (for calls/SMS)

4) Line and phone status check

5) Gate led and gate opening

6) SMS/call led

7) Phone line status led

8) Reset button polling

9) Expiring users check

```
GateO $ Common Led_Managing Phone_book SIM800 SMS Serial _GO_Defs.h _GO_GlobVar.h _GO_Lang_EN.h _GO_Lang_IT.h _GO_TestMsg.h
88
89
90
91 void loop() {
92   #ifndef _TEST_NO_WDT
93     wdt_reset(); //reset watchdog
94   #endif
95   #ifdef _TEST_ON_USB
96     SerialRead(); // monitors USB serial port and manage commands received
97   #endif
98   iTime = micros();
99   while (CheckSIM800Requests()) { //monitors messages received by SIM800 (phone calls, SMS, status messages, ...)
100     iTime = micros();
101   };
102   SetTiming(micros() - iTime, ALERT); // SetTiming stores time spent (avg and max) for each loop phase.
103   if ((smsStatus == 'Q') && (!SIM800.available())){ //if there are no pending activities on SIM800 go on checking line and SIM800
104     iTime = micros();
105     GsmCsq(); //check signal quality, SIM800 and simcard
106     SetTiming(micros() - iTime, CSQ);
107   }
108   iTime = micros();
109   GateComm(); //monitors gate relay and gate led
110   SetTiming(micros() - iTime, GATE);
111   iTime = micros();
112   SmsLed(); //drives calls/sms led
113   SetTiming(micros() - iTime, S_LED);
114   iTime = micros();
115   GsmLed(); //drives line status led
116   SetTiming(micros() - iTime, G_LED);
117   iTime = micros();
118   CheckHandReset(); //monitors reset button
119   SetTiming(micros() - iTime, RES_B);
120   if (smsStatus == 'Q') //if there are no pending activities on SIM800
121     CleanSimCard(); //monitors sim card phonebook cleaning for expired registrations
122   oLoopTime = loopTime;
123   loopTime = micros();
124   SetTiming(loopTime - oLoopTime, LOOP);
125 }
```

Sketch organization: monitoring timing

```
103  if (SMSStatus == Q) && (!SIM000.available()) {
104      iTime = micros();
105      GsmCsq();
106      SetTiming(micros() - iTime, CSQ);
107  }
```

Sketch organization: loop()

- 1) Watchdog timer reset
- 2) USB polling/writing
- 3) Phone polling (for calls/SMS)
- 4) Line and phone status check
- 5) Gate led and gate opening
- 6) SMS/call led
- 7) phone/card/line status led
- 8) Reset button polling
- 9) Expiring users check

```
GateO $ Common Led_Managing Phone_book SIM800 SMS Serial _GO_Defs.h _GO_GlobVar.h _GO_Lang_EN.h _GO_Lang_IT.h _GO_TestMsg.h
88
89
90
91 void loop() {
92   #ifndef _TEST_NO_WDT
93     wdt_reset(); //reset watchdog
94   #endif
95   #ifdef _TEST_ON_USB
96     SerialRead(); // monitors USB serial port and manage commands received
97   #endif
98   iTime = micros();
99   while (CheckSIM800Requests()) { //monitors messages received by SIM800 (phone calls, SMS, status messages, ...)
100     iTime = micros();
101   };
102   SetTiming(micros() - iTime, ALERT); // SetTiming stores time spent (avg and max) for each loop phase.
103   if ((smsStatus == 'Q') && (!SIM800.available())){ //if there are no pending activities on SIM800 go on checking line and SIM800
104     iTime = micros();
105     GsmCsq(); //check signal quality, SIM800 and simcard
106     SetTiming(micros() - iTime, CSQ);
107   }
108   iTime = micros();
109   GateComm(); //monitors gate relay and gate led
110   SetTiming(micros() - iTime, GATE);
111   iTime = micros();
112   SmsLed(); //drives calls/sms led
113   SetTiming(micros() - iTime, S_LED);
114   iTime = micros();
115   GsmLed(); //drives line status led
116   SetTiming(micros() - iTime, G_LED);
117   iTime = micros();
118   CheckHandReset(); //monitors reset button
119   SetTiming(micros() - iTime, RES_B);
120   if (smsStatus == 'Q') //if there are no pending activities on SIM800
121     CleanSimCard(); //monitors sim card phonebook cleaning for expired registrations
122   oLoopTime = loopTime;
123   loopTime = micros();
124   SetTiming(loopTime - oLoopTime, LOOP);
125 }
```



“

TIMING ACTIVITIES on ARDUINO:
managing millis() and micros()

”

Next video