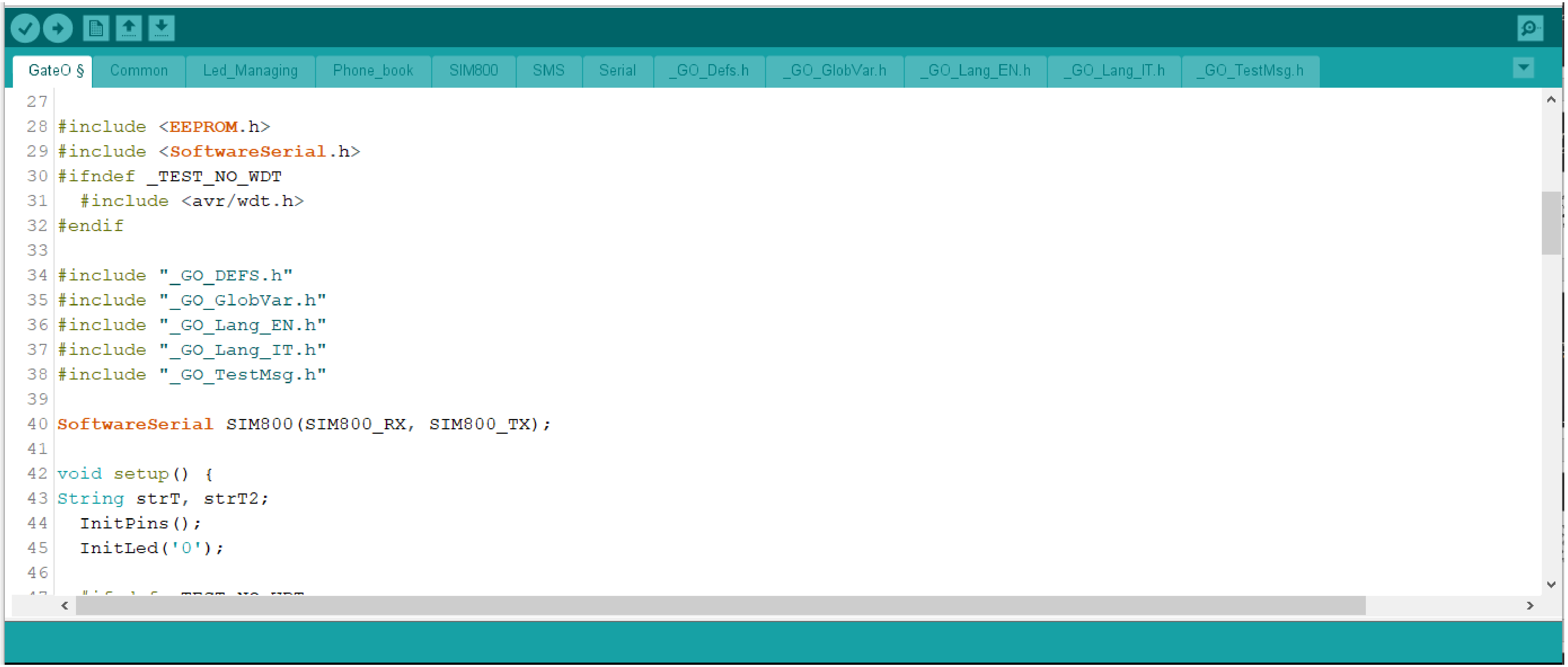
ARDUINO + SIM800L

LO SKETCH

```
  } else if (*comm==F("TIMEZONE")){
    test=(((param->length())==3) && (String("+-").indexOf(param->charAt(0))!=-1) && (isDigit(param->charAt(1))) && (
    if (te t && (param->toInt()>(-48)) && (param->toInt()<(49))) {
      f   k=0; k<3; k++)
          ROM.update(EE_TZ_ADDR+k,param->charAt(k));
      s     =F(_M_SM045);
            =F(_M_SM012);
            y=strT1+strT2;
      k     ROM.read(EE_NTP_ADDR);
      i     k==0) || (k==255)){ // user is setting TIMEZON
            T=F(EE_DEF_NTP_SERV);
            (k=0; k<(strT.length());k++)
          EEPROM.update(EE_NTP_ADDR+k,strT.charAt(k));
        EEPROM.update(EE_NTP_ADDR+(strT.length()),ch
      }
      return true;
    } else {
      *reply=F(_M_SM008);
      *reply+=String(char(10));
      *reply+=F(_M_SM009);
      return false;
    }

  } else if (*comm==F("NTP")){
    if (*param=="") {
      strT1=F(_M_SM085);
```

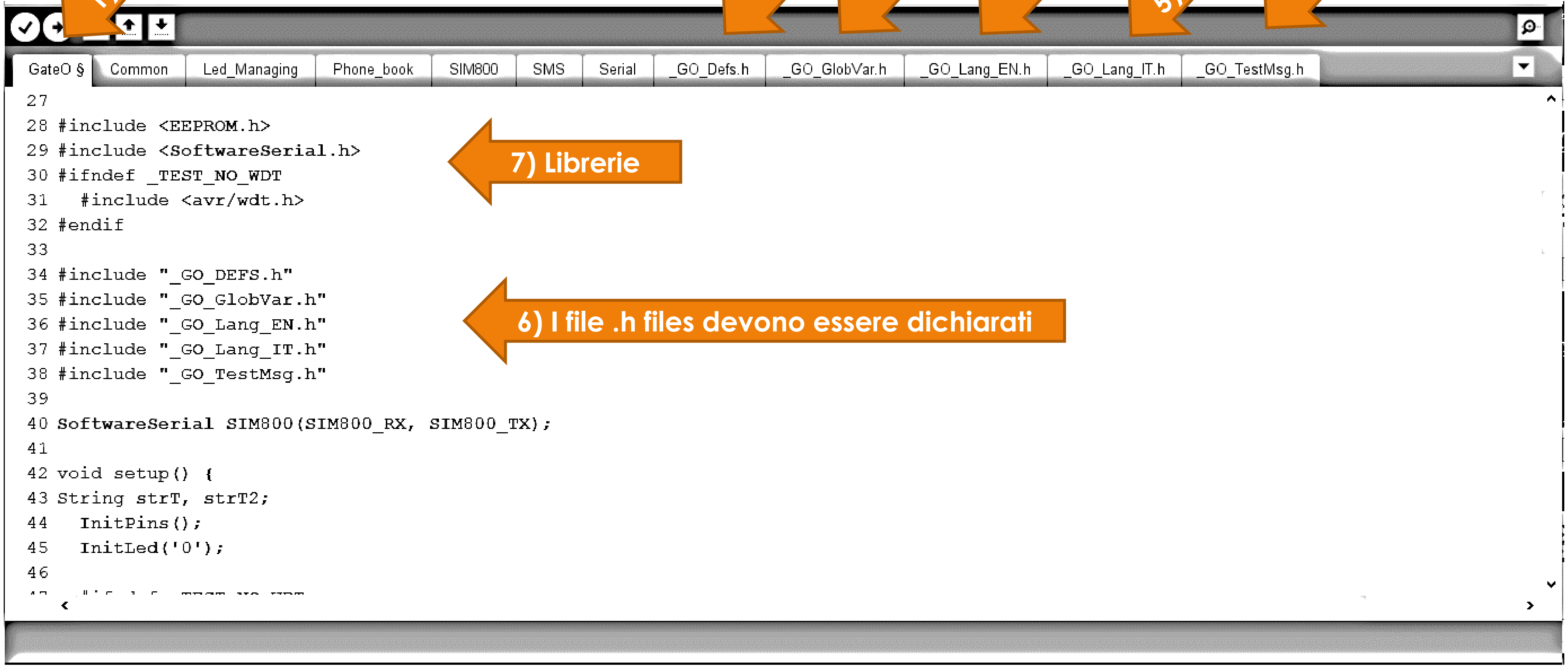
L'APRICANCELLO GSM

Organizzazione dello sketch



```

#include <EEPROM.h>
#include <SoftwareSerial.h>
#ifndef _TEST_NO_WDT
  #include <avr/wdt.h>
#endif

#include "_GO_DEFS.h"
#include "_GO_GlobVar.h"
#include "_GO_Lang_EN.h"
#include "_GO_Lang_IT.h"
#include "_GO_TestMsg.h"

SoftwareSerial SIM800(SIM800_RX, SIM800_TX);

void setup() {
String strT, strT2;
  InitPins();
  InitLed('0');

  #ifndef _TEST_NO_WDT
```

Organizzazione dello sketch



1) Sketch Principale

2) #Define

3) Variablili globali

5) Messaggi multilingua

5) Messaggi multilingua

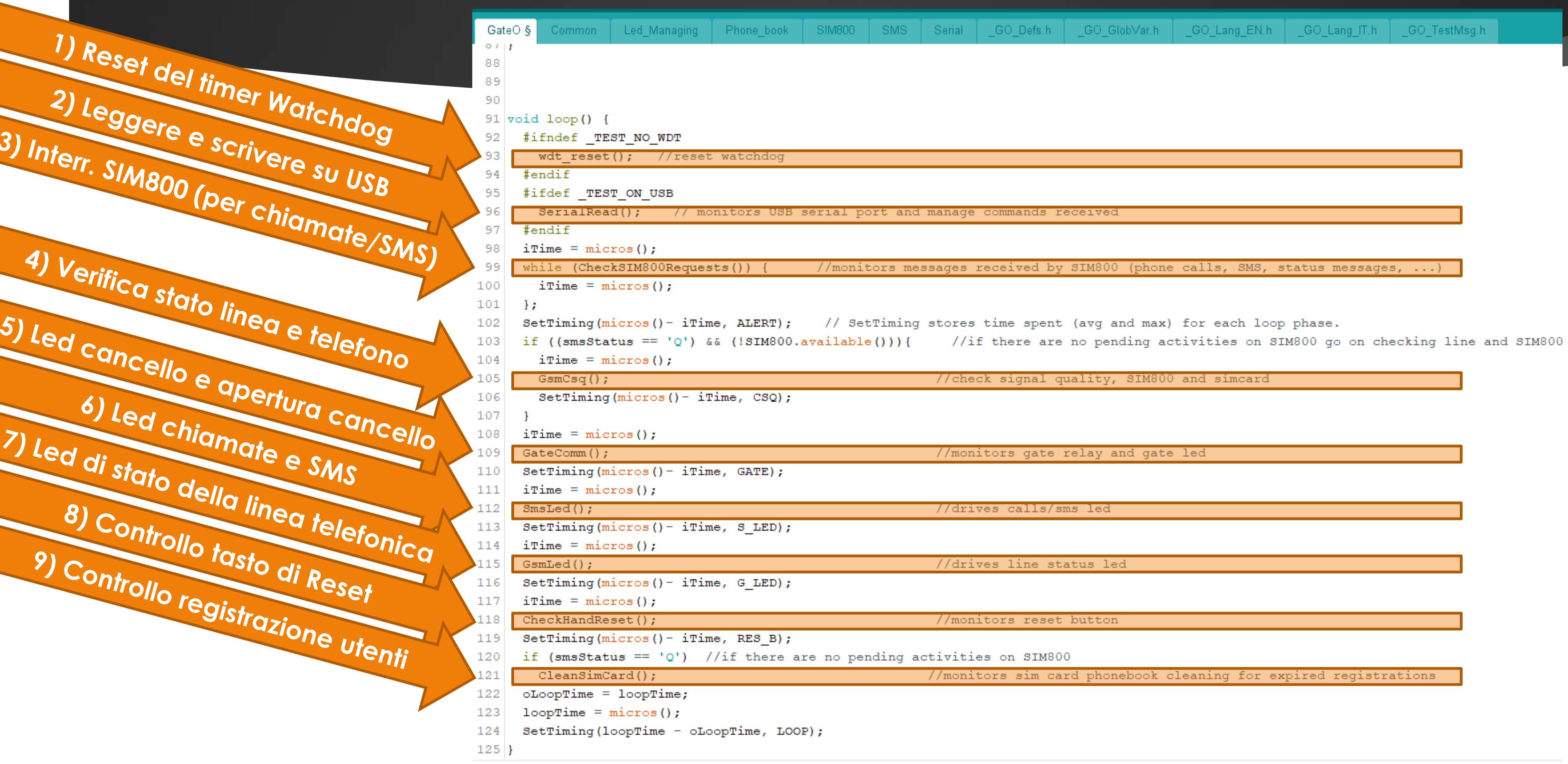
4) Macro per Debug

```

#include <EEPROM.h>
#include <SoftwareSerial.h>
#ifndef _TEST_NO_WDT
  #include <avr/wdt.h>
#endif

#include "_GO_DEFS.h"
#include "_GO_GlobVar.h"
#include "_GO_Lang_EN.h"
#include "_GO_Lang_IT.h"
#include "_GO_TestMsg.h"

SoftwareSerial SIM800(SIM800_RX, SIM800_TX);

void setup() {
String strT, strT2;
  InitPins();
  InitLed('0');

  #ifndef _TEST_NO_WDT
```

7) Librerie

6) I file .h files devono essere dichiarati

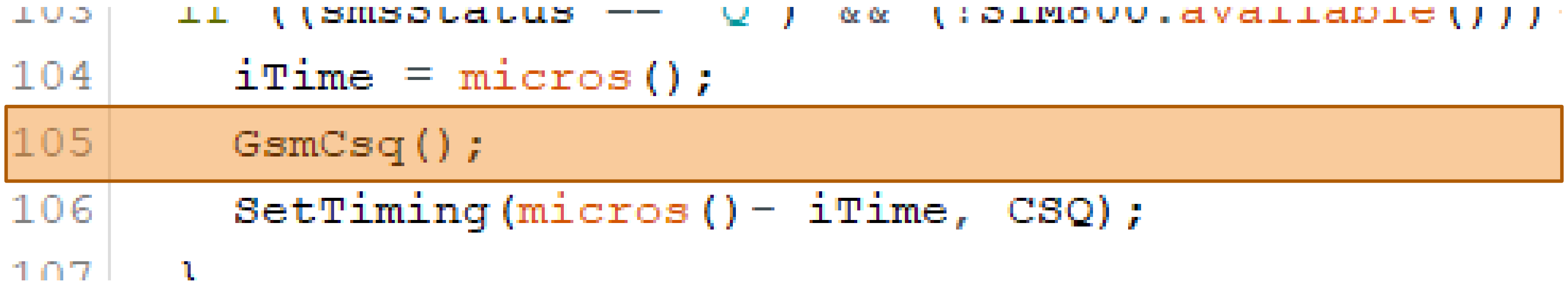
Organizzazione dello sketch: loop()



1) Reset del timer Watchdog
2) Leggere e scrivere su USB
3) Interr. SIM800 (per chiamate/SMS)
4) Verifica stato linea e telefono
5) Led cancello e apertura cancello
6) Led chiamate e SMS
7) Led di stato della linea telefonica
8) Controllo tasto di Reset
9) Controllo registrazione utenti

```
}



void loop() {
  #ifndef _TEST_NO_WDT
    wdt_reset();   //reset watchdog
  #endif
  #ifdef _TEST_ON_USB
    SerialRead();     // monitors USB serial port and manage commands received
  #endif
  iTime = micros();
  while (CheckSIM800Requests()) {       //monitors messages received by SIM800 (phone calls, SMS, status messages, ...)
    iTime = micros();
  };
  SetTiming(micros()- iTime, ALERT);    // SetTiming stores time spent (avg and max) for each loop phase.
  if ((smsStatus == 'Q') && (!SIM800.available())){      //if there are no pending activities on SIM800 go on checking line and SIM800
    iTime = micros();
    GsmCsq();                                             //check signal quality, SIM800 and simcard
    SetTiming(micros()- iTime, CSQ);
  }
  iTime = micros();
  GateComm();                                             //monitors gate relay and gate led
  SetTiming(micros()- iTime, GATE);
  iTime = micros();
  SmsLed();                                               //drives calls/sms led
  SetTiming(micros()- iTime, S_LED);
  iTime = micros();
  GsmLed();                                               //drives line status led
  SetTiming(micros()- iTime, G_LED);
  iTime = micros();
  CheckHandReset();                                       //monitors reset button
  SetTiming(micros()- iTime, RES_B);
  if (smsStatus == 'Q')  //if there are no pending activities on SIM800
    CleanSimCard();                                      //monitors sim card phonebook cleaning for expired registrations
  oLoopTime = loopTime;
  loopTime = micros();
  SetTiming(loopTime - oLoopTime, LOOP);
}
```

Organizzazione dello sketch: rilevare i tempi



```
  if ((smsStatus == 'Q') && (!SIM800.available()))
    iTime = micros();
    GsmCsq();
    SetTiming(micros()- iTime, CSQ);
  }
```

Organizzazione dello sketch: loop()

1) Reset del timer Watchdog

2) Leggere e scrivere su USB

3) Interr. SIM800 (per chiamate/SMS)

4) Verifica stato linea e telefono

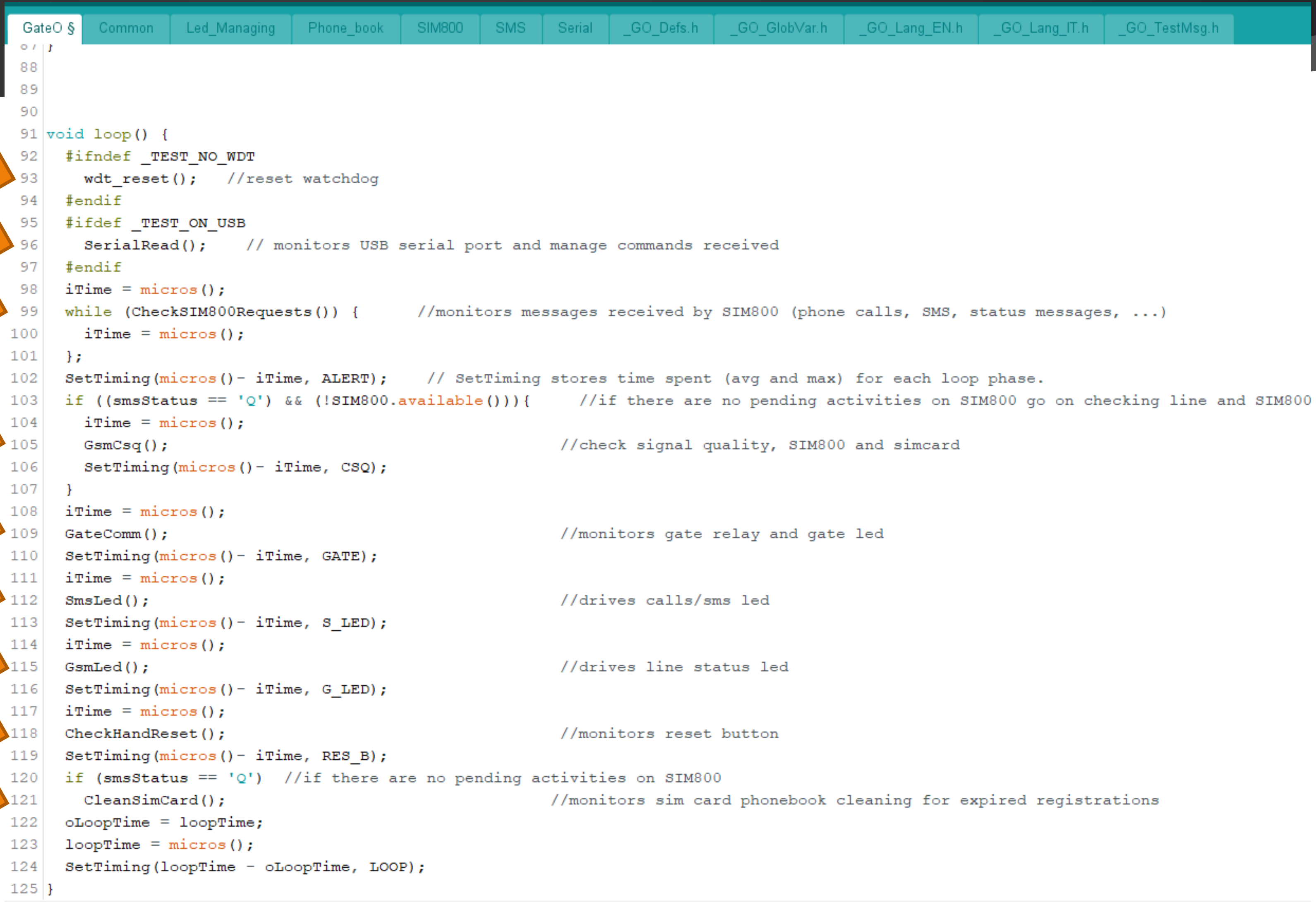
5) Led cancello e apertura cancello

6) Led chiamate e SMS

7) Led di stato della linea telefonica

8) Controllo tasto di Reset

9) Controllo registrazione utenti



```
}



void loop() {
  #ifndef _TEST_NO_WDT
    wdt_reset();   //reset watchdog
  #endif
  #ifdef _TEST_ON_USB
    SerialRead();    // monitors USB serial port and manage commands received
  #endif
  iTime = micros();
  while (CheckSIM800Requests()) {      //monitors messages received by SIM800 (phone calls, SMS, status messages, ...)
    iTime = micros();
  };
  SetTiming(micros()- iTime, ALERT);    // SetTiming stores time spent (avg and max) for each loop phase.
  if ((smsStatus == 'Q') && (!SIM800.available())){     //if there are no pending activities on SIM800 go on checking line and SIM800
    iTime = micros();
    GsmCsq();                                         //check signal quality, SIM800 and simcard
    SetTiming(micros()- iTime, CSQ);
  }
  iTime = micros();
  GateComm();                                         //monitors gate relay and gate led
  SetTiming(micros()- iTime, GATE);
  iTime = micros();
  SmsLed();                                           //drives calls/sms led
  SetTiming(micros()- iTime, S_LED);
  iTime = micros();
  GsmLed();                                           //drives line status led
  SetTiming(micros()- iTime, G_LED);
  iTime = micros();
  CheckHandReset();                                   //monitors reset button
  SetTiming(micros()- iTime, RES_B);
  if (smsStatus == 'Q')  //if there are no pending activities on SIM800
    CleanSimCard();                                  //monitors sim card phonebook cleaning for expired registrations
  oLoopTime = loopTime;
  loopTime = micros();
  SetTiming(loopTime - oLoopTime, LOOP);
}
```

“

TEMPORIZZARE le attività di ARDUINO: gestire millis() e micros()

”

Next video

Prossimo video