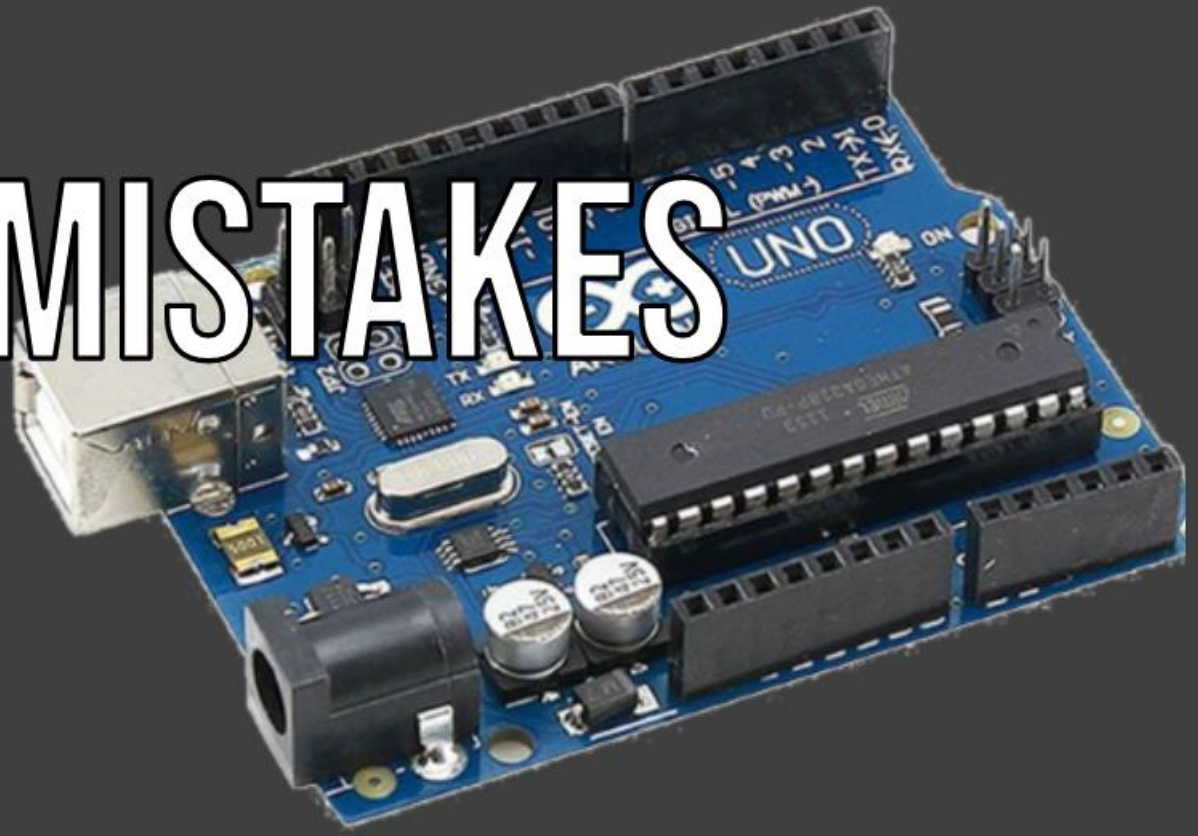
MILLIS() - MICROS()

AND TIMING MISTAKES



ARDUINO

Organizzazione dello sketch: loop()

1) Reset del timer Watchdog
2) scrittura/lettura porta USB
3) Controllo se chiamate/SMS
4) Verifica stato linea e hw
5) Apertura cancello
6) Gestione led SMS/chiamate
7) Gestione led di stato linea
8) Controllo pulsante Reset
9) Controllo registraz. scadute

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 }
```

Organizzazione dello sketch: controllo disponibilità linea GSM

4) Verifica stato della linea e del telefono

7) Gestione led di stato linea/tel.

```
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 g
        iTime = micros();
        GsmCsq();                                          //check signal quality, SIM800 and simcard
        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
      SetTiming(micros()- iTime, S_LED);
      iTime = micros();
      GsmLed();                                            //drives line status led
116   SetTiming(micros()- iTime, G_LED);
117   iTime = micros();
```

GsmCsq + GsmLed:

CONTROLLO:

SIM800, simcard, linea GSM

AGGIORNAMENTO:

led di stato

RESET:

di SIM800 in caso di errore

- GsmCsq()
  - controlla: SIM800 attivo e funzionante; simcard attiva; livello segnale GSM
  - Controllato dal timer GSM_CSQ_TIMING
  - Imposta la variabile globale: «gsmStatus»
- GsmLed()
  - Legge la variabile globale «gsmStatus» e di conseguenza…
  - Controlla il led GSM che mostra lo stato del telefono e della linea
  - E' controllato dai timer:
    - LED_LONG_DELAY
    - LED_FLASH_ON
    - LED_FLASH_OFF
  - Se «gsmStatus» é «F»=guasto ed il timer SIM800_DELAY_BEFORE_RESET è scaduto, avvia il reset di SIM800

MILLIS()
MICROS()

Restituisce «unsigned long» (da 0 a 4,294,967,295)

Logica "complemento a 2":

4,294,967,295 + 1 = 0

Millis() si azzera ogni 50 giorni

Micros() si azzera ogni 72 minuti

Logica "complemento a 2":

4,294,967,250 + 70 = 24

24 - 4,294,967,250 = 70

MILLIS()
MICROS()

Usare millis(), micros()
e attività temporizzate

- **CASE A**

```
#define TIMER 1000
unsigned long nextTime=0;
(loop())
    If (millis()>=nextTime) {
        … (attività temporizzata) …
        nextTime = millis()+TIMER;
    }
```

- **CASE B**

```
#define TIMER 1000
unsigned long prevTime=0;
(loop())
    If ((millis()-prevTime)>=TIMER) {
        … (attività temporizzata) .
        prevTime = millis();
    }
```