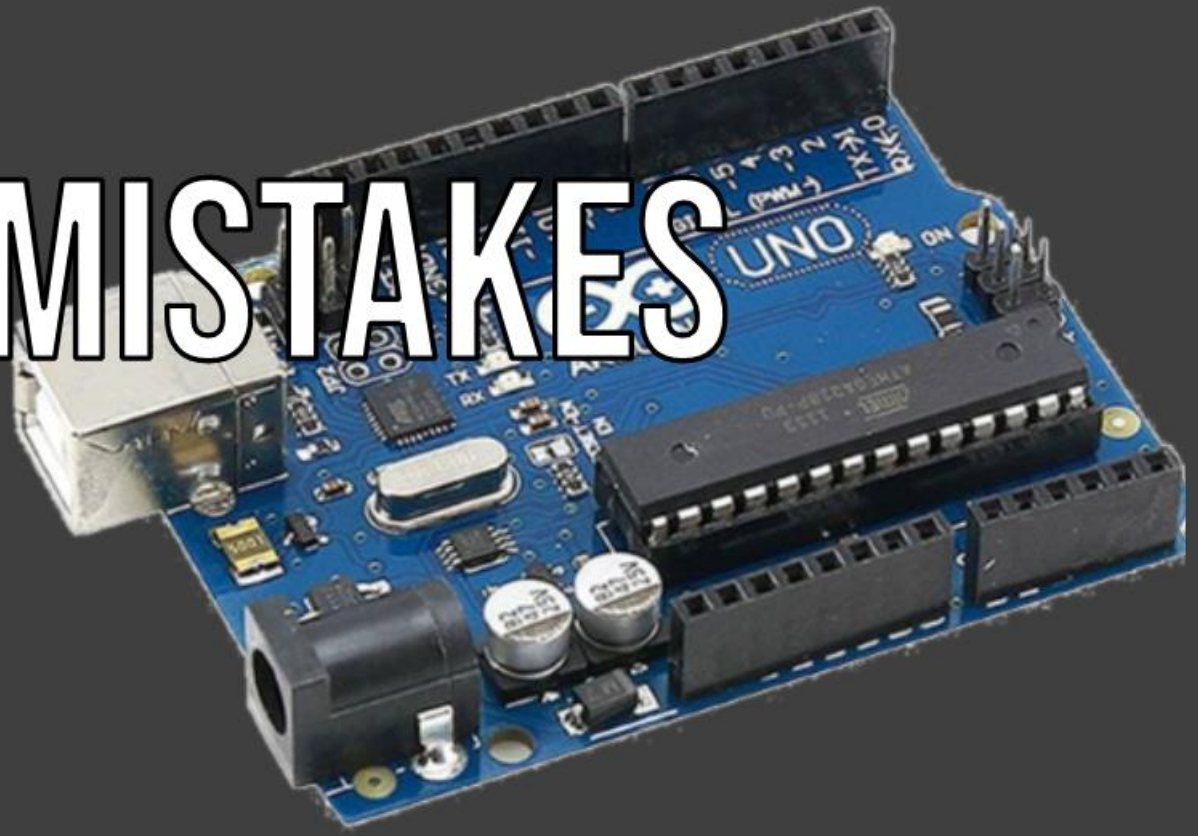
MILLIS() - MICROS()

AND TIMING MISTAKES



ARDUINO

Sketch organization: loop()

1) Watchdog timer reset
2) USB polling/writing
3) Phone polling (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

```
}


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);
}
```

Sketch organization: GSM network availability control

4) Line and phone status check

7) phone/card/line status led

```
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:

MONITORING:

SIM800, simcard, linea GSM

UPDATING:

status led

RESET:

SIM800 reset in case of failure

- GsmCsq()
  - monitoring: SIM800 is ON and working; simcard enabled; GSM signal level
  - Timing reference: GSM_CSQ_TIMING
  - It sets the global variable: «gsmStatus»
- GsmLed()
  - Refer to «gsmStatus» global variable and consequently…
  - Monitor GSM led, showing phone/line statud
  - Timing references:
    - LED_LONG_DELAY
    - LED_FLASH_ON
    - LED_FLASH_OFF
  - If «gsmStatus» is set to «F»=fault and timer SIM800_DELAY_BEFORE_RESET expired, it executes SIM800 reset

MILLIS()
MICROS()

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

“Two’s complement” logic

4,294,967,295 + 1 = 0

Millis() resets every 50 giorni

Micros() resets every 72 minuti

“Two’s complement” logic

4,294,967,200 + 100 = 5

5 - 4,294,967,200 = 100

MILLIS()
MICROS()

How to use millis() and micros() in timed activities

- **CASE A**

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

- **CASE B**

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