Arduino playground : **Code** / **Metro**

```
Metro library for Arduino
by Thomas Ouellet Fredericks
contact: mrtoftrash@gmail.com
```

CURRENT VERSION

2.3 2009/01/19 Added an autoreset argument so I can use either the original check() behavior or benjamin.soelberg's suggested one (see below)

HISTORY

2.2 2008/09/23 Updated examples 2.1 2008/09/19 Updated documentation 1.0 2006/06/27 Initial Release

DESCRIPTION

Metro is a library for Arduino (arduino.cc).

It facilitates the implementation of recurring timed events like:
- blinking LEDs
- servo motor control
- Serial communication

HOW TO IMPORT/INSTALL

Download here: Metro version 2.2

Put the Metro folder in "hardware\libraries\".

In the Arduino IDE, create a new sketch (or open one) and

select from the menubar "Sketch->Import Library->Metro".

Once the library is imported, an "#inlcude Metro.h" line will appear

at the top of your Sketch.

CREATION

Metro(unsigned long interval)

Instantiates a Metro object with a set interval in milliseconds (autoreset = false).

Metro(unsigned long interval, byte autoreset)

Instantiates a Metro object with a set interval in milliseconds.

If the autoreset is set to true (1), the internal timer will reset will reset, ignoring missed events. If you want to catch up with missed events (because you don't call the check method regularly), set autoreset to false.

METHODS

Because Metro does not use interrupts, you have to "check" the Metro regularly (at least every millisecond).

byte check()

Returns true if the interval has lapsed. Returns false if not.

void interval(unsigned long interval)

Changes the interval in milliseconds.

void reset()

Restarts/resets the Metro.

EXAMPLE 1: Blinking a led

```
/*
 This code will blink an LED attached to pin 13 on and off.
 It will stay on for 0.25 seconds.
 It will stay off for 1 second.
 */
#include <Metro.h> //Include Metro library
#define LED 13 // Define the led's pin

//Create a variable to hold theled's current state
int state = HIGH;

// Instantiate a metro object and set the interval to 250 milliseconds (0.25 seconds).
Metro ledMetro = Metro(250);

void setup()
{
  pinMode(LED,OUTPUT);
  digitalWrite(LED,state);
}

void loop()
```

```
{

  if (ledMetro.check() == 1) { // check if the metro has passed it's interval .
    if (state==HIGH)  {
      state=LOW;
      ledMetro.interval(250); // if the pin is HIGH, set the interval to 0.25 seconds.
    }
    else {
      ledMetro.interval(1000); // if the pin is LOW, set the interval to 1 second.
      state=HIGH;
    }
    digitalWrite(LED,state);
  }
}
```

EXAMPLE 2: Serial Interval

```
// This example sends a Serial message every 250 milliseconds

#include <Metro.h> // Include the Metro library

Metro serialMetro = Metro(250);  // Instantiate an instance

void setup() {
  Serial.begin(115200); // Start the Serial communication
}

void loop() {

  if (serialMetro.check() == 1) { // check if the metro has passed it's interval .
  // Output all the analog readings seperated by a space character
    for (int i = 0; i < 6; i++ ) {
      Serial.print (analogRead( i) );
      Serial.print(32,BYTE);
    }
    // Terminate message with a linefeed and a carriage return
    Serial.print(13,BYTE);
    Serial.print(10,BYTE);
  }
}
```

(Printable View of http://www.arduino.cc/playground/Code/Metro)