Arduino library TimedAction

History

Array Library for Arduino
Author: Alexander Brevig
Contact: alexanderbrevig@gmail.com

1.0 2009-03-23 - Alexander Brevig : Initial Release
1.1 2009-04-08 - Alexander Brevig : Added an example that demonstrates three arduino examples at once
1.2 2009-04-13 - Alexander Brevig : Added a constructor
1.3 2009-04-16 - Alexander Brevig : Added disable() and enable(), requested by: http://www.arduino.cc/cgi-bin/yabb2/YaBB.pl?action=viewprofile;username=ryno
1.4 2009-05-06 - Alexander Brevig : Added reset()
1.5 2009-10-25 - Alexander Brevig : Added setInterval , requested by: Boris Neumann
1.6 2010-10-08 - Alexander Brevig : Changed datatype of interval from unsigned int to unsigned long
2.0 2019-11-20 - Marcel Jordaan : Corrected version arduino version >= 100

Description

TimedAction is a library for the Arduino.

It is created to help hide the mechanics of how to implement Protothreading and general millis() timing. It is sutied for those actions that needs to happen approximately every x milliseconds.

Download, install and import

Library ZIP file: TimedAction.zip

Add the utility library, in the Arduino IDE select from the menubar "Sketch->Include Library->Add .ZIP Library". Select the library file 'TimedAction.zip' and confirm the choice, library will now be added.

In the Arduino IDE, create a new sketch (or open one) and select from the menubar "Sketch->Import Library->TimedAction".

Once the library is imported, an '#include <TimedAction.h>' line will appear at the top of your Sketch.

Creation

```
TimedAction(unsigned int interval,void (*function)())
TimedAction(unsigned long preDelay,unsigned int interval,void (*function)
())

TimedAction timedAction = TimedAction(1000,blink);
```

Instanciates a TimedAction object that will trigger void blink() every second, if checked.

Note: The preDelay might be useful if you want to make sure that you have a connection to, for instance an network server, before the TimedAction starts triggering.

Methods

void check()

Will execute the function it is referred to. In this case, the void blink(), if interval has been met.

void enable()

Will enable "thread" execution.

void disable()

Will disable "thread" execution.

void reset()

Will reset the time relative to the TimedAction.

void setInterval(unsigned int newInterval)

Change the interval of this TimedAction.

Example

BlinkAction

```
#include <TimedAction.h>

//this initializes a TimedAction class that will change the state of an LED
every second.
TimedAction timedAction = TimedAction(1000,blink);

//pin / state variables
#define ledPin 13
boolean ledState = false;


void setup(){
  pinMode(ledPin,OUTPUT);
  digitalWrite(ledPin,ledState);
}

void loop(){
  timedAction.check();
}

void blink(){
  ledState ? ledState=false : ledState=true;
  digitalWrite(ledPin,ledState);
}
```

ThreeExamplesAtOnce

```
/*
||
|| @file ThreeExamplesAtOnce.pde
```

```
|| @version 1.0
|| @author Alexander Brevig
|| @contact alexanderbrevig@gmail.com
||
|| @description
|| | This sketch blinks an LED as Blink
|| |             sets a led on or off according to serial buffer as
PhysicalPixel
|| |             prints the ascii table as ASCIITable
|| #
||
*/

#include <TimedAction.h>

//this initializes a TimedAction object that will change the state of an
LED every second.
TimedAction blinkAction                 =       TimedAction(1000,blink);
//this initializes a TimedAction object that will change the state of an
LED
//according to the serial buffer contents, every 50 milliseconds
TimedAction physicalPixelAction =       TimedAction(50,physicalPixel);
//this initializes a TimedAction object that will write tha ascii table to
the serial every ten seconds
TimedAction asciiTableAction    =       TimedAction(10000,asciiTable);

//pin / state variables
#define ledPin 13
#define physicalPin 12
boolean ledState = false;


void setup(){
  pinMode(ledPin,OUTPUT);
  digitalWrite(ledPin,ledState);
  pinMode(physicalPin, OUTPUT);
  Serial.begin(9600);
}

void loop(){
  blinkAction.check(); //trigger every second
  physicalPixelAction.check(); //trigger every 50 millisecond
  asciiTableAction.check(); //trigger every 10 second
}
```

```
//[url=https://arduino.cc/en/Tutorial/Blink]Examples->Digital->Blink[/url]
void blink(){
  ledState ? ledState=false : ledState=true;
  digitalWrite(ledPin,ledState);
}

//[url=https://arduino.cc/en/Tutorial/PhysicalPixel]Examples->Digital-
>PhysicalPixel[/url]
void physicalPixel()
{
  if (Serial.available()) {
    byte val = Serial.read();
    if (val == 'H') {
      digitalWrite(physicalPin, HIGH);
    }
    if (val == 'L') {
      digitalWrite(physicalPin, LOW);
    }
  }
}

//[url=https://arduino.cc/en/Tutorial/ASCIITable]Examples->Digital-
>ASCIITable[/url]
void asciiTable()
{
  byte number = 33; // first visible character '!' is #33
  // print until we have printed last visible character '~' #126 ...
  while(number <= 126) {
    Serial.print(number, BYTE);    // prints value unaltered, first will be
'!'

    Serial.print(", dec: ");
    Serial.print(number);          // prints value as string in decimal
(base 10)
    // Serial.print(number, DEC);  // this also works

    Serial.print(", hex: ");
    Serial.print(number, HEX);     // prints value as string in hexadecimal
(base 16)

    Serial.print(", oct: ");
    Serial.print(number, OCT);     // prints value as string in octal (base
8)

    Serial.print(", bin: ");
```

```
    Serial.println(number, BIN);   // prints value as string in binary
(base 2)
                                                                    //
also prints ending line break
    number++; // to the next character
  }
  asciiTableAction.disable();
}
```

FAQ

> How can I use multiple TimedActions?

TimedAction is a class. Therefore to make multiple TimedActions you must create an instance for each of them. In the example BlinkAction above, a TimedAction instance (TimedAction timedAction) is bound to the void blink() function, and set to trigger every second, if checked.

```
TimedAction timedAction = TimedAction(1000,blink);
```

To add a TimedAction to an additional function (void updateLcd() or instance) and set to trigger every 200 millisecond, you could create the following instance timedAction2:

```
TimedAction timedAction2 = TimedAction(NO_PREDELAY,200,updateLcd);
```

And now it's just to keep updating those two instances:

```
//update instances and possibly fire funcitons
timedAction1.check();
timedAction2.check();
```

See ThreeExamplesAtOnce for an example on how to use multiple TimedActions.

> How can I use long intervals?

For timers longer than about a minute, you will need to change the type of all variable and parameter definitions for intervl or interval from unsigned int to unsigned long (2 in the .cpp and 3 in the .h).