wdt_samd21 Library

A very simple library to activate, reset and deactivate the WDT on ATSAMD21.

Based on the work of MartinL (https://forum.arduino.cc/u/MartinL) on Arduino forum (Apr, 2018)

The wdt_samd21 library allows, in a very easy way, on ATSAMD21 MCU, to **activate**, to periodically **reset**, to **deactivate** and to **reactivate** the WDT (*Watch Dog Timer*), which are the normal required functions for a simple use of the WDT for checking the correct execution of an application program.

Library usage and initialization

Customization

To define the "*timeout*" of the WDT you can use the constants defined in the SAMD21 core, in the wdt.h file:

```
WDT_CONFIG_PER_8        8 clock cycles ( 7.8 msec.)
WDT_CONFIG_PER_16      16 clock cycles (15.6 msec.)
WDT_CONFIG_PER_32      32 clock cycles (31.2 msec.)
WDT_CONFIG_PER_64      64 clock cycles (62.5 msec.)
WDT_CONFIG_PER_128    128 clock cycles ( 125 msec.)
WDT_CONFIG_PER_256    256 clock cycles ( 250 msec.)
WDT_CONFIG_PER_512    512 clock cycles ( 500 msec.)
WDT_CONFIG_PER_1K    1024 clock cycles ( 1 sec.)
WDT_CONFIG_PER_2K    2048 clock cycles ( 2 sec.)
WDT_CONFIG_PER_4K    4096 clock cycles ( 4 sec.)
WDT_CONFIG_PER_8K    8192 clock cycles ( 8 sec.)
WDT_CONFIG_PER_16K  16384 clock cycles (16 sec.)
```

... the defaul value, if nothing is passed to the initialization function, is `WDT_CONFIG_PER_2K` .

Initialization

To use this library first you have to add, at the beginning of your program:

```
#include <wdt_samd21.h>
```

... next you have to call the library functions.

Library functions

wdt_init(unsigned long wdt_config_per)

Initialize the WDT with a timeout equal to the value passed as a parameter. It **must be** one of the values described in the "Customization" paragraph.

Example:

```
wdt_init ( WDT_CONFIG_PER_1K );
```

wdt_reset( )

Must be called before the *timeout* time passes to reset the WDT counter. If you do not call it in time, the MCU **reset**.

Example:

```
wdt_reset ( );
```

wdt_disable( )

Disable the WDT until it is reactivated again with a wdt_reEnable( ).

Example:

```
wdt_disable ( );
```

wdt_reEnable( )

Re-enable the WDT disabled by a previous wdt_disable( ).

Example:

```
wdt_reEnable ( );
```

Demo Program

The following example initializes the WDT for a timeout of 2 seconds, after which, in the loop(), it performs a for structure with a delay() of one second at each iteration, but sending a wdt_reset() command to the WDT to avoid the restart. At the end of the for structure, the wdt is first disabled then, after a 3 second delay(), is enabled again and, finally, a 4 second delay() is performed which causes the MCU to restart so, all that following the last delay(), is never executed.

```
#include <wdt_samd21.h>

void setup() {
   delay ( 500 );
   //
   Serial.begin ( 9600 );
   while ( !Serial ) {
      delay ( 100 );
   }
   //
   // Initialze WDT with a 2 sec. timeout
   wdt_init ( WDT_CONFIG_PER_2K );
}

void loop() {
   for ( byte i = 0; i < 5; i++ ) {
      // wait a second
      delay ( 1000 );
      // write on the serial port
      Serial.print ( "Iteration " );
      Serial.print ( i + 1 );
      Serial.println ( " of 5" );
      // "feed" the WDT to avoid restart
      wdt_reset();
   }
   //
   // now disable wdt and wait ...
   wdt_disable();
   Serial.println( "wdt disabled ..." );
   Serial.println ( "Now waiting for 3 seconds ..." );
   delay(3000);
   //
   // ... then reEnable the wdt ...
   wdt_reEnable();
   Serial.println( "wdt reEnabled ..." );
   //
   // ... and wait 4 seconds ... the WDT should restart the board
   Serial.println ( "Now waiting for 4 seconds ..." );
   delay ( 4000 );
   //
   Serial.println ( "*** You will never see this message printed ***"
);
   delay ( 1000 );
}
```