25C_KernelPeriod_test

Using the new Node `KernelPeriod', the kernel period can be changed by just modifying the Node parameter.

```
/* ================================================================ *
 *                Automatically generated by n-Blocks Studio 2.0    *
 *                                                                  *
 *                          www.n-blocks.net                        *
 * ================================================================ */
#include "nlib\nblocks.h"
#include "nlib\BSP\bsp.h"
// Custom nodes:
#include "nlib\Ticker\ticker.h"
#include "nlib\StringFormat\stringformat.h"
#include "nlib\StringSerial\stringserial.h"
#include "nlib\KernelPeriod\kernelperiod.h"

// -*-*- List of node objects -*-*-
nBlock_Ticker           nb_nBlockNode0_Ticker      (1000);
nBlock_StringFormat     nb_nBlockNode1_StringFormat ("Value: %d!\n");
nBlock_StringSerial     nb_nBlockNode2_StringSerial (USBTX, USBRX);

nBlock_KernelPeriod     nb_nBlocksNode3_nBlocks_KernelPeriod(0.5);

// -*-*- List of connection objects -*-*-
nBlockConnection    n_conn0( &nb_nBlockNode0_Ticker, 0,    &nb_nBlockNode1_StringFormat,    0);
nBlockConnection    n_conn1( &nb_nBlockNode1_StringFormat, 0,    &nb_nBlockNode2_StringSerial,    0);


// -*-*- Main function -*-*-
int main(void) {
    SetupWorkbench();
    while(1) {
        ProgressNodes();

        // Your custom code here!
    }
}

```

This nBlocksStudio project sends "Value: 1\n" to the serial port every second. Is a working project to test the new Kernel version `[a14d29e]` kernel data broadcast, fixed designs without connections

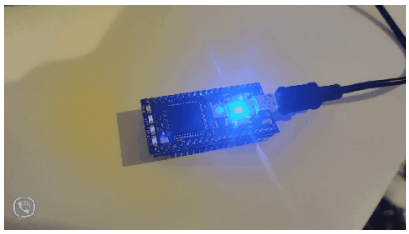
The 'normal' kernel tick period is 0.001 sec. Here the Node `KernelPeriod` set it up to 0.5s and still the kernel outputs all text correctly to the serial port

Kernel blink Led

```
11  DigitalOut led1(LED1);
195         if (__propagating == 0) {
196             __propagating = 1; // Flag: we are in the middle of a frame
197             led1 = !led1;
```

For this testing and until a new Node KernelBlink is created, the Kernel is modified (only locally in this example directory, in the file workbench.cpp) to toggle an LED in every tick.

Testing



- Led blinks at twice the tick period ✔
- The string "Value: 1\n" is printed in the serial port ✔
- The printing happens every 1000ms ✔
- Compiled and Build with mbed-studio os5 bare-metal ✔

```
Link: 25C_KernelPeriod_test
Elf2Bin: 25C_KernelPeriod_test
Post-Build: 25C_KernelPeriod_test
| Module             |          .text |       .data |        .bss |
|--------------------|----------------|-------------|--------------|
| [fill]             |      602(+602) |       4(+4) |     27(+27) |
| [lib]\c.a          | 31004(+31004) | 2472(+2472) |     89(+89) |
| [lib]\gcc.a        |    4064(+4064) |       0(+0) |       0(+0) |
| [lib]\misc         |     180(+180) |       4(+4) |     28(+28) |
| main.o             |     382(+382) |       0(+0) |    880(+880) |
| mbed-os\drivers    |    1874(+1874) |       0(+0) |       0(+0) |
| mbed-os\hal        |    1932(+1932) |       4(+4) |     67(+67) |
| mbed-os\platform   |    5484(+5484) |    260(+260) |    364(+364) |
| mbed-os\targets    |    3336(+3336) |       4(+4) |    241(+241) |
| nlib\KernelPeriod  |       44(+44) |       0(+0) |       0(+0) |
| nlib\StringFormat  |     220(+220) |       0(+0) |       0(+0) |
| nlib\StringSerial  |     144(+144) |       0(+0) |       0(+0) |
| nlib\Ticker        |     318(+318) |       0(+0) |       0(+0) |
| nlib\nworkbench.o  |     788(+788) |      12(+12) |    112(+112) |
| Subtotals          | 50372(+50372) | 2760(+2760) | 1808(+1808) |
Total Static RAM memory (data + bss): 4568(+4568) bytes
Total Flash memory (text + data): 53132(+53132) bytes
Image: BUILD/LPC1768/GCC_ARM\25C_KernelPeriod_test.bin
```

Node KernelPeriod source code notes

```
#include "kernelperiod.h"


nBlock_KernelPeriod::nBlock_KernelPeriod(float period) {
    KernelPeriod(period);
    KernelTickSource(KERNEL_TICK_TIMER, NC);
}
```

The first parameter is the source, can be either `KERNEL_TICK_TIMER` or `KERNEL_TICK_EXT`.

If `KERNEL_TICK_EXT` is used, then the second argument is the pin name for the external clock to be used as tick