

SCHEMATIC_10D_MotorTest

Schematic Design project to test the new Nodes:

- [Node L298](#) ✓
- [Node GPITrigger](#) ✕
- [Node ConstTrigger](#) ✕
- [Node ValueTrig](#) ✓

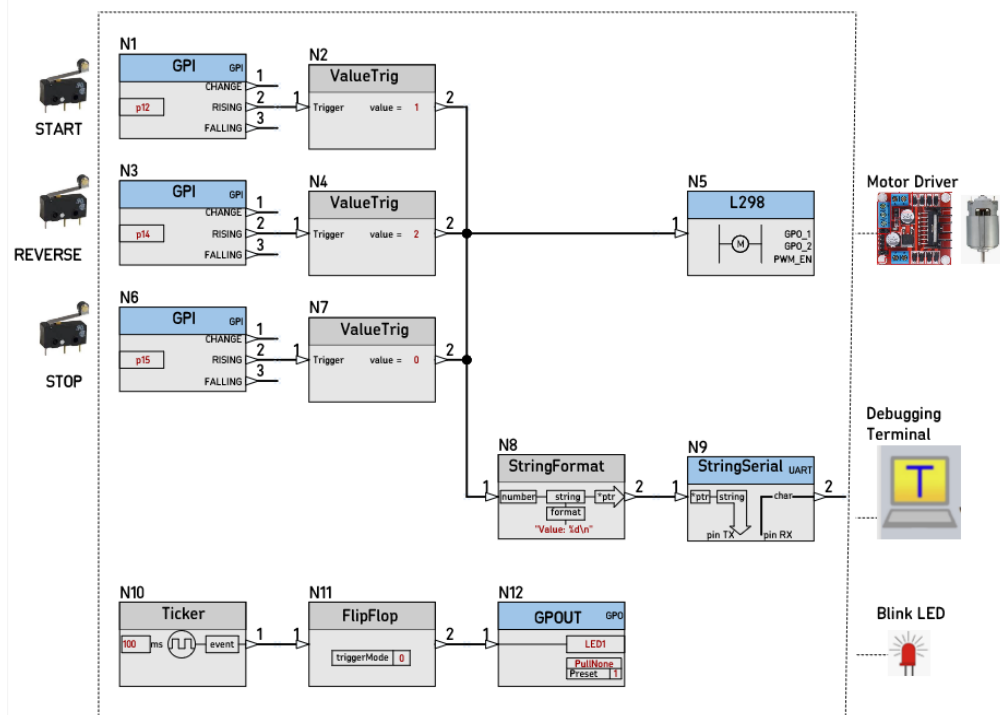
To be automatically translated from nBlocksStudio Translator.

The C++ project is here: [10D_MotorTest](#)

Use case:

- Button [Transition 0 to 3.3V] at Pin p12 starts the movement, direction right
- EndSwitch [Transition 0 to 3.3V] at Pin p14 change direction
- EndSwitch [Transition 0 to 3.3V] at Pin p15 stop

Schematic nBlocksStudio Design ✓



Translation Pass ✓

```
F:\n-blocks\10D_MotorTest\studio>translate.cmd
F:\n-blocks\10D_MotorTest\studio>echo off
F:\n-blocks\10D_MotorTest\studio>F:\n-blocks\studio\studio2.0_experimental\python\python.exe F:\n-blocks\studio\studio2.0_experimental\nblocksStudio.py --import altium --netlist .\studio.NET --parameters .\studio.csv --output .\export --name 10D_MotorTest --verbose
=====
| n-Blocks Studio |
=====
--- Welcome to n-Blocks Studio development environment ---

Studio working location: F:\n-blocks\studio\studio2.0_experimental

Offline database loaded - starting n-Blocks Studio
Importing Altium design...
Generating export output...
Preparing required libraries...
  Checking: GPI
    Content is up-to-date
  Checking: ValueTrig
    Content is up-to-date
  Checking: L298
    Content is up-to-date
  Checking: StringFormat
    Content is up-to-date
  Checking: StringSerial
    Content is up-to-date
  Checking: Ticker
    Content is up-to-date
  Checking: FlipFlop
    Content is up-to-date
  Checking: GPOUT
    Content is up-to-date
    Board support package already present, current version preserved
Embedded code exported to ".\export\10D_MotorTest"

- Thank you for using n-Blocks Studio ---
-----
F:\n-blocks\10D_MotorTest\studio>
```

```

main.cpp x
1
2  /* ===== *
3   *      Automatically generated by n-Blocks Studio 2.0      *
4   *      *
5   *      www.n-blocks.net      *
6   *      ===== */
7  #include "nlib\nblocks.h"
8  #include "nlib\BSP\bsp.h"
9  // Custom nodes:
10 #include "nlib\GPI\gpi.h"
11 #include "nlib\ValueTrig\valuetrig.h"
12 #include "nlib\L298\L298.h"
13 #include "nlib\StringFormat\stringformat.h"
14 #include "nlib\StringSerial\stringserial.h"
15 #include "nlib\Ticker\ticker.h"
16 #include "nlib\FlipFlop\flipflop.h"
17 #include "nlib\GPOUT\gpout.h"
18
19 // -*- List of node objects -*-
20 nBlock_GPI          nb_nBlockNode0_GPI          (p12);
21 nBlock_ValueTrig    nb_nBlockNode1_ValueTrig    (1);
22 nBlock_GPI          nb_nBlockNode2_GPI          (p14);
23 nBlock_ValueTrig    nb_nBlockNode3_ValueTrig    (2);
24 nBlock_L298         nb_nBlockNode4_L298         (LED2, LED3, LED4);
25 nBlock_GPI          nb_nBlockNode5_GPI          (p15);
26 nBlock_ValueTrig    nb_nBlockNode6_ValueTrig    (0);
27 nBlock_StringFormat nb_nBlockNode7_StringFormat ("Value: %d\n");
28 nBlock_StringSerial nb_nBlockNode8_StringSerial (USBTX, USBRX);
29 nBlock_Ticker        nb_nBlockNode9_Ticker      (100);
30 nBlock_FlipFlop      nb_nBlockNode10_FlipFlop   (0);
31 nBlock_GPOUT         nb_nBlockNode11_GPOUT      (LED1, PullNone, 1);
32
33 // -*- List of connection objects -*-
34 nBlockConnection    n_conn0( &nb_nBlockNode10_FlipFlop, 0, &nb_nBlockNode11_GPOUT, 0);
35 nBlockConnection    n_conn1( &nb_nBlockNode9_Ticker, 0, &nb_nBlockNode10_FlipFlop, 0);
36 nBlockConnection    n_conn2( &nb_nBlockNode7_StringFormat, 0, &nb_nBlockNode8_StringSerial, 0);
37 nBlockConnection    n_conn3( &nb_nBlockNode5_GPI, 1, &nb_nBlockNode6_ValueTrig, 0);
38 nBlockConnection    n_conn4( &nb_nBlockNode2_GPI, 1, &nb_nBlockNode3_ValueTrig, 0);
39 nBlockConnection    n_conn5( &nb_nBlockNode1_ValueTrig, 0, &nb_nBlockNode4_L298, 0);
40 nBlockConnection    n_conn6( &nb_nBlockNode1_ValueTrig, 0, &nb_nBlockNode7_StringFormat, 0);
41 nBlockConnection    n_conn7( &nb_nBlockNode3_ValueTrig, 0, &nb_nBlockNode4_L298, 0);
42 nBlockConnection    n_conn8( &nb_nBlockNode3_ValueTrig, 0, &nb_nBlockNode7_StringFormat, 0);
43 nBlockConnection    n_conn9( &nb_nBlockNode6_ValueTrig, 0, &nb_nBlockNode4_L298, 0);
44 nBlockConnection    n_conn10( &nb_nBlockNode6_ValueTrig, 0, &nb_nBlockNode7_StringFormat, 0);
45 nBlockConnection    n_conn11( &nb_nBlockNode0_GPI, 1, &nb_nBlockNode1_ValueTrig, 0);
46
47
48 // -*- Main function -*-
49 int main(void) {
50     SetupWorkbench();
51     while(1) {
52         ProgressNodes();
53
54         // Your custom code here!
55     }
56 }
57

```

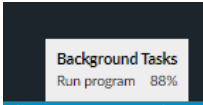
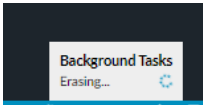
Compiler Pass ✓

```
Building project 100_MotorTest (LPC1768, GCC_ARM)
Scan: 100_MotorTest
Link: 100_MotorTest
Elf2Bin: 100_MotorTest
Post-Build: 100_MotorTest
```

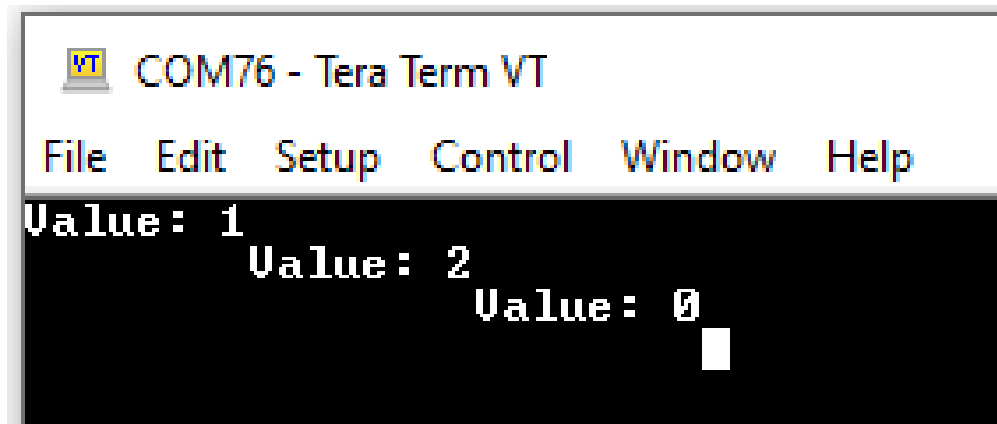
Module	.text	.data	.bss
[fill]	610(+0)	4(+0)	27(+0)
[lib]\c.a	31004(+0)	2472(+0)	89(+0)
[lib]\gcc.a	4980(+0)	0(+0)	0(+0)
[lib]\misc	180(+0)	4(+0)	28(+0)
main.o	742(+0)	0(+0)	1572(+0)
mbed-os\drivers	2118(+0)	0(+0)	0(+0)
mbed-os\hal	1932(+0)	4(+0)	67(+0)
mbed-os\platform	5484(+0)	260(+0)	364(+0)
mbed-os\targets	3714(+0)	32(+0)	245(+0)
nlib\FlipFlop	160(+0)	0(+0)	0(+0)
nlib\GPI	546(+0)	0(+0)	0(+0)
nlib\GPOUT	196(+0)	0(+0)	0(+0)
nlib\L298	544(+0)	0(+0)	0(+0)
nlib\StringFormat	214(+0)	0(+0)	0(+0)
nlib\StringSerial	144(+0)	0(+0)	0(+0)
nlib\Ticker	318(+0)	0(+0)	0(+0)
nlib\ValueTrig	90(+0)	0(+0)	0(+0)
nlib\nworkbench.o	560(+0)	0(+0)	88(+0)
Subtotals	53536(+0)	2776(+0)	2480(+0)

```
Total Static RAM memory (data + bss): 5256(+0) bytes
Total Flash memory (text + data): 56312(+0) bytes
Image: BUILD/LPC1768/GCC_ARM\100_MotorTest.bin
```

Flash Programming Pass ✓



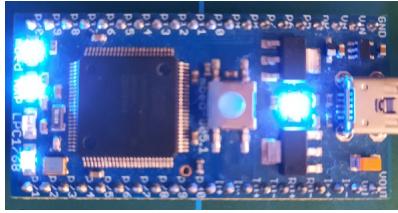
Serial Terminal Pass ✓



The screenshot shows a terminal window titled "COM76 - Tera Term VT". The menu bar includes "File", "Edit", "Setup", "Control", "Window", and "Help". The terminal output consists of three lines: "Value: 1", "Value: 2", and "Value: 0", with a cursor at the end of the third line.

```
COM76 - Tera Term VT
File Edit Setup Control Window Help
Value: 1
      Value: 2
            Value: 0
```

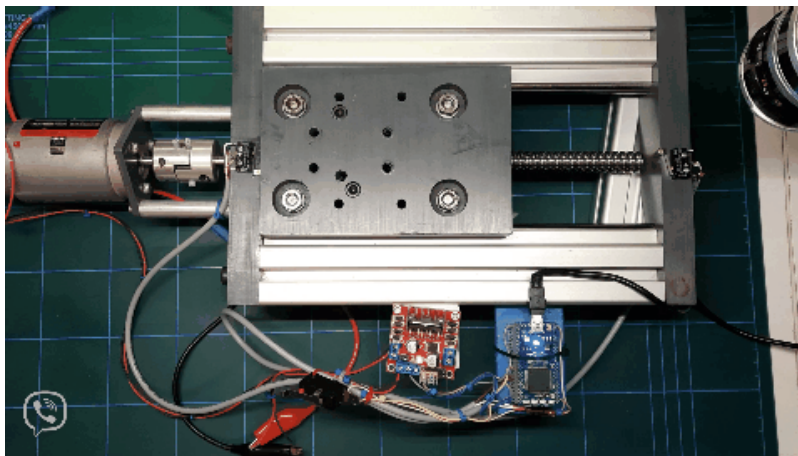
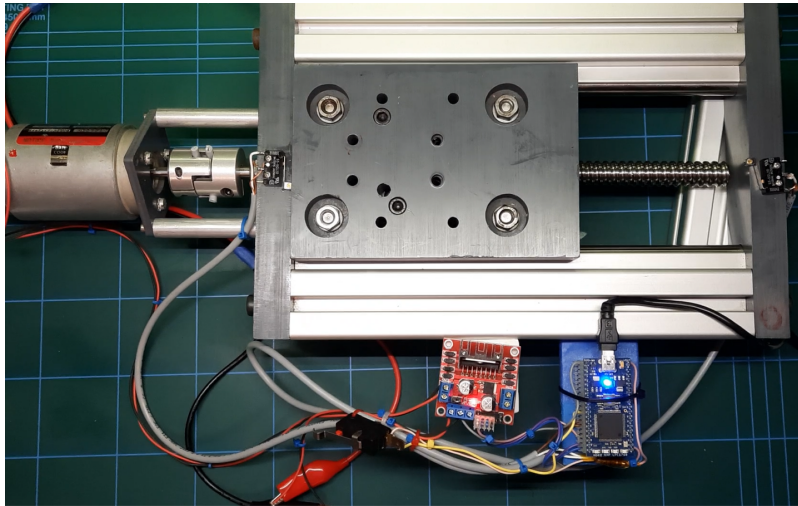

Led testing Pass ✓



- * LED1: Is the Blink LED
- * LED2 = 1, LED3 = 0 (L298 IN2, IN2): Movement Direction Right
- * LED 4 = Low Duty Cycle 5%: Motor is turning Right

```
44         default:
45             stop();
46             break;
47         } //switch
48     } //if(must_update)
49 } //endFrame
50
51 void nBlock_L298::stop(void) {
52     _in1 = OFF;
53     _in2 = OFF;
54 }
55
56 void nBlock_L298::turnLeft(void) {
57     _enable.write(0.95f);
58     _in1 = OFF;
59     _in2 = ON;
60 }
61
62 void nBlock_L298::turnRight(void) {
63     _enable.write(0.05f);
64     _in1 = ON;
65     _in2 = OFF;
66 }
67
68 void nBlock_L298::brake(void) {
69     _enable.write(0.95f);
70     _in1 = ON;
71     _in2 = ON;
72 }
73
```

Next: test with LPC1768-mbed, L298-breakout and a DC Motor ?



Next: Test with Nucleo STM32F401 AND Nucleo DC motor expansion board ?