An introduction to firmware development for power electronics

by Remco Nijenhuis

POWER ELECTRONICS 2022

Power Electronics & Energy Storage event

14 juni 2022 | 1931 Congrescentrum 's-Hertogenbosch

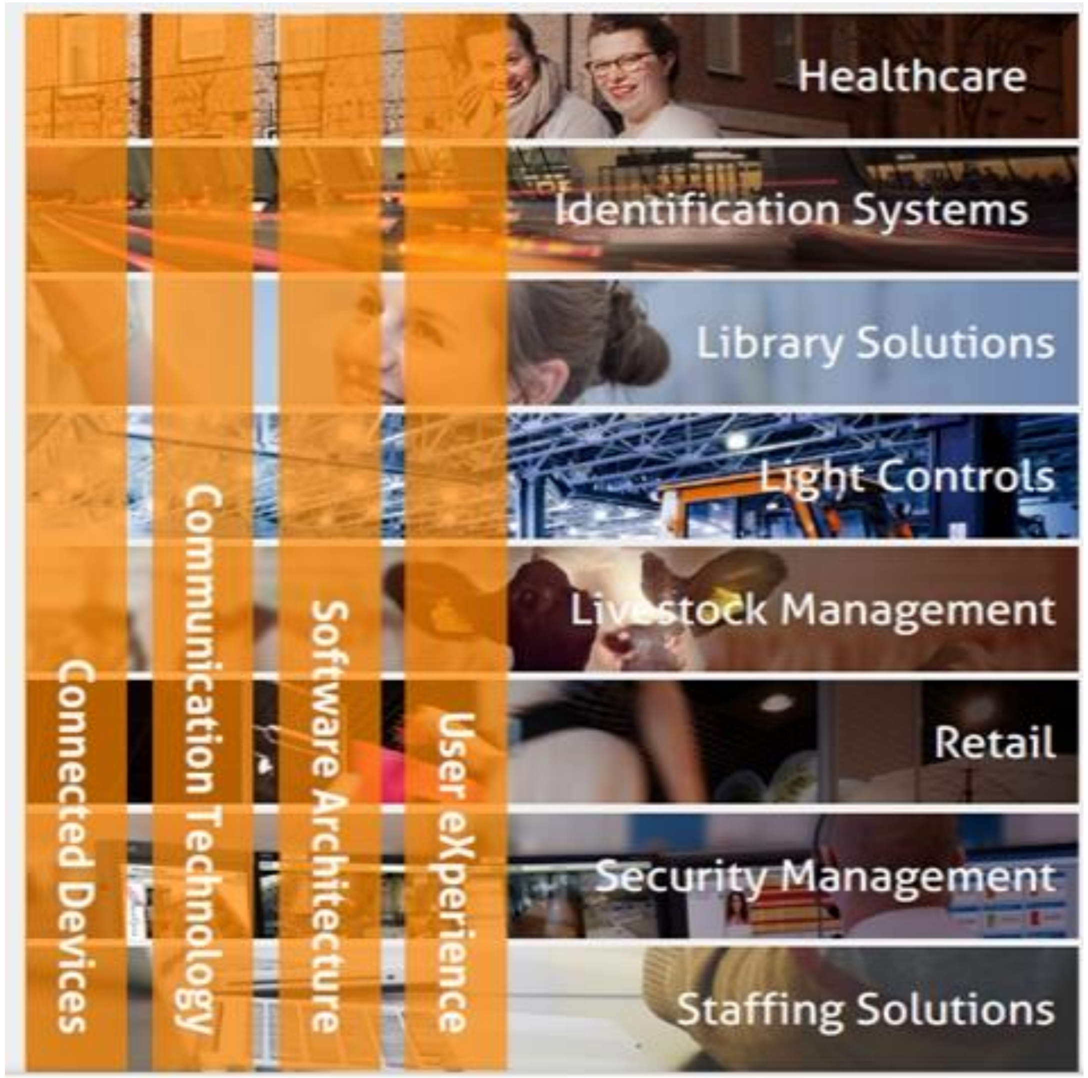
ENERGY STORAGE EVENT 2022

Power Electronics & Energy Storage event

POWER ELECTRONICS 2022 ENERGY STORAGE EVENT 2022

14 juni 2022 | 1931 Congrescentrum 's-Hertogenbosch

Nedap core technologie fields



nedap

UV lamp drivers

- Lamp drivers from 100W up to 36kW
- Up to 4000 Volts ignition peak voltage
- Up to 1200 Volts continuous voltage
- Up to 30 Ampere continuous current

4x220W

6x4kW rack

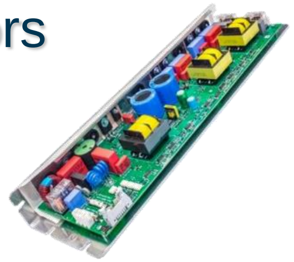
12kW

24kW

36kW

nedap

New York - 56 reactors
12.000 Lamps
2.9MW



Chicago
Waste Water plant

1200 million liters/day

nedap

Introduction Remco Nijenhuis

- Electronics engineer
- Firmware architect / engineer

- RFID
- Airfield lighting
- Automotive
- UV lamp drivers

Firmware development: Safety First

Firmware development: Safety First

No developer → No code → No product

Firmware development: Safety First

Use development boards or scaled models:

During firmware development do not use the actual high power hardware when it is not needed.

Firmware development: Safety First

Hardware safe by design:

When the CPU is not running the device must be safe by hardware measures

nedap

Firmware development: Safety First

Use galvanic isolators:

During development on the actual hardware use a galvanic isolator for protection, like an isolated variac or transformer.

nedap

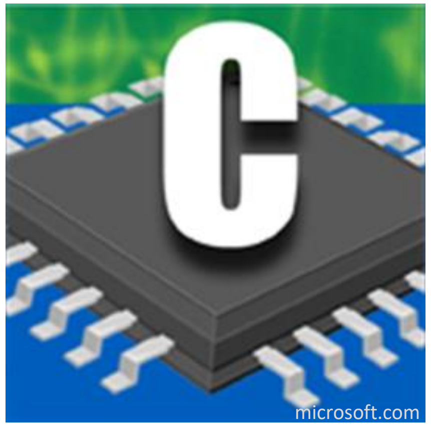
Firmware development: Safety First

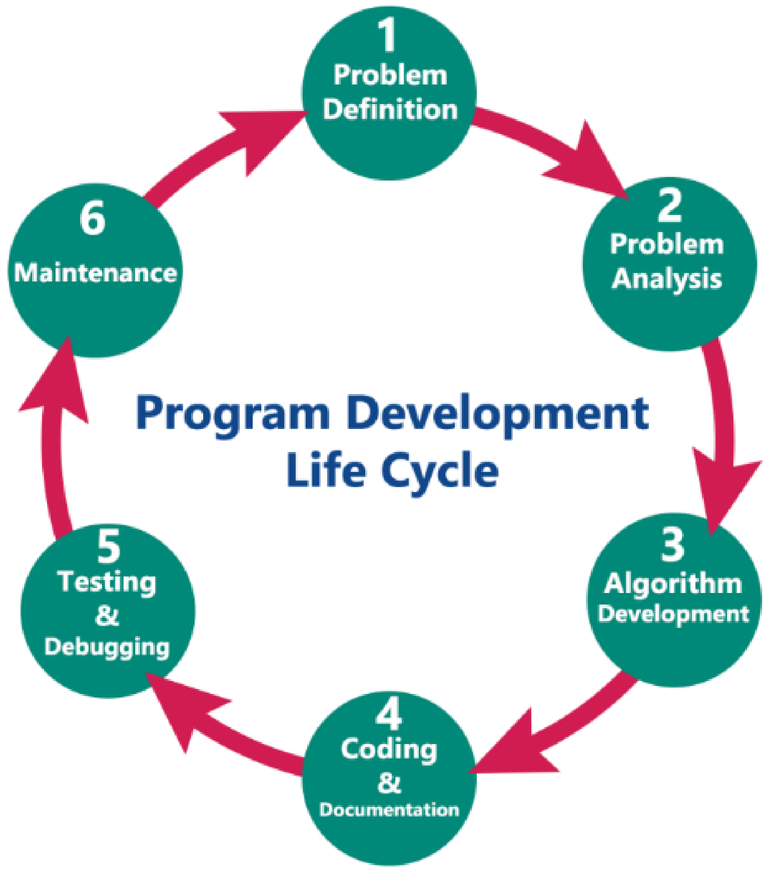
Usage of debuggers, emulators or programmers:

Be sure what the effect of using these devices can be on your hardware during programming or debugging.

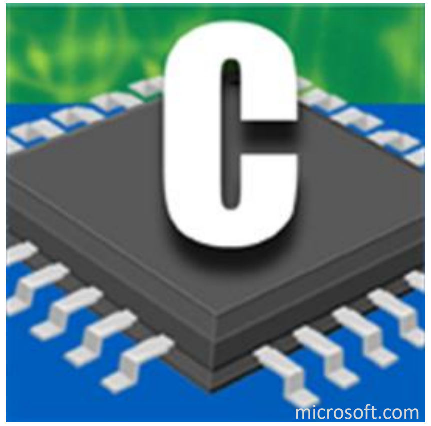
- Shared pinning of debuggers and actual hardware
- Using breakpoints
- Isolator between hardware and debugger

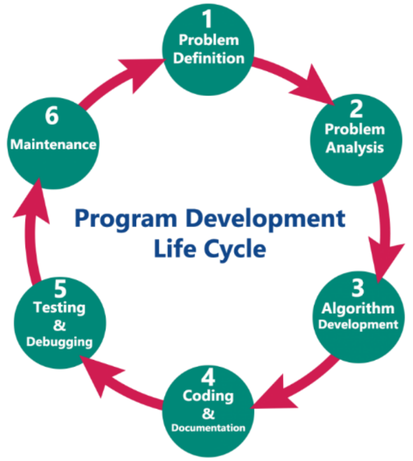
Firmware development: Writing code





Firmware development: Writing code

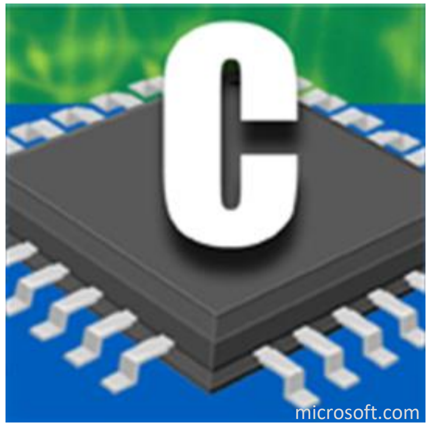




Use processes during firmware development:

- Use a versioning system (svn, git)
- Use a ticket system (Jira)
- Workflows keeps you organized
- Forces you to take the same steps everytime.

Firmware development: Writing code

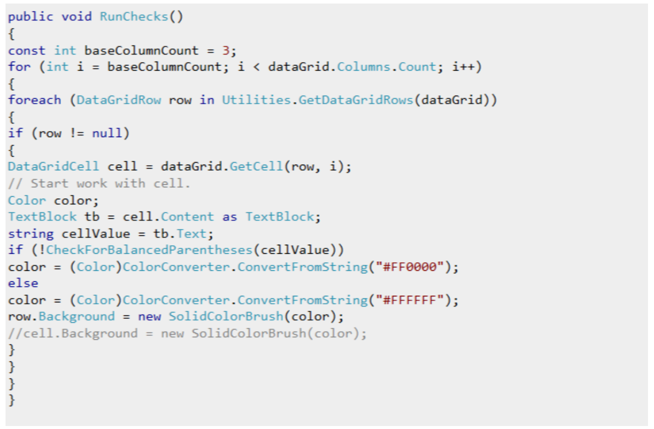


What is the quality of your code?

- How to prove the quality of your code.
- Continuous improvement

nedap

Firmware development: Code Quality

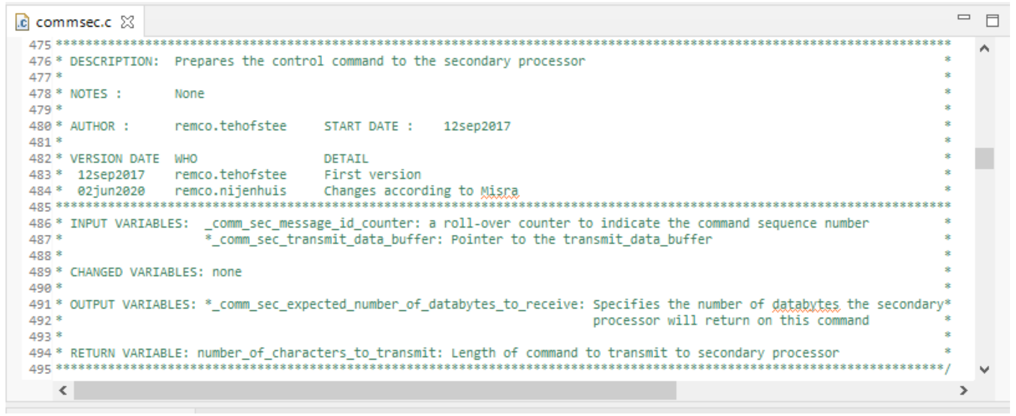


```
public void RunChecks()
{
const int baseColumnCount = 3;
for (int i = baseColumnCount; i < dataGrid.Columns.Count; i++)
{
foreach (DataGridRow row in Utilities.GetDataGridRows(dataGrid))
{
if (row != null)
{
DataGridCell cell = dataGrid.GetCell(row, i);
// Start work with cell.
Color color;
TextBlock tb = cell.Content as TextBlock;
string cellValue = tb.Text;
if (!CheckForBalancedParentheses(cellValue))
color = (Color)ColorConverter.ConvertFromString("#FF0000");
else
color = (Color)ColorConverter.ConvertFromString("#FFFFFF");
row.Background = new SolidColorBrush(color);
//cell.Background = new SolidColorBrush(color);
}
}
}
}
```

Use a coding style and a code formatter:

- Improve the readability of your code
- Programming clear code is not only for yourself

Firmware development: Code Quality



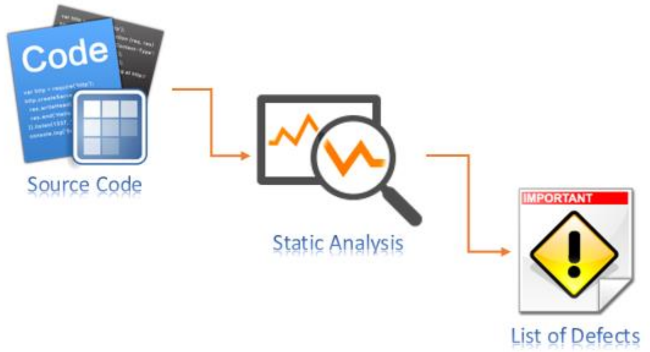
```
***************************************************************************************************************************
* DESCRIPTION:  Prepares the control command to the secondary processor                                                  *
*                                                                                                                         *
* NOTES :       None                                                                                                      *
*                                                                                                                         *
* AUTHOR :      remco.tehofstee     START DATE :    12sep2017                                                             *
*                                                                                                                         *
* VERSION DATE  WHO                 DETAIL                                                                                *
*  12sep2017    remco.tehofstee     First version                                                                         *
*  02jun2020    remco.nijenhuis     Changes according to Misra                                                            *
***************************************************************************************************************************
* INPUT VARIABLES:  _comm_sec_message_id_counter: a roll-over counter to indicate the command sequence number             *
*                   *_comm_sec_transmit_data_buffer: Pointer to the transmit_data_buffer                                  *
*                                                                                                                         *
* CHANGED VARIABLES: none                                                                                                 *
*                                                                                                                         *
* OUTPUT VARIABLES: *_comm_sec_expected_number_of_databytes_to_receive: Specifies the number of databytes the secondary*
*                                                                       processor will return on this command             *
*                                                                                                                         *
* RETURN VARIABLE: number_of_characters_to_transmit: Length of command to transmit to secondary processor                 *
**************************************************************************************************************************/
```

Use clear names and add comment to your code:

- Use proper names for variables and constants.
- A few lines of comment can help a lot in the future.

Firmware development: Code Quality

techwell.com



cyberhoot.com

Use a static code analyzer:

- It as an objective analysis of your code always according the same rules.

nedap

Firmware development: Code Quality

techwell.com

```
client = new bl.deskto
tionClient.Insert();

cationClient.LastRequest = DateTi
cationClient.RequestCount = Notifi
ficationClient.Update();

onClient.Deny == false)
equest Notification
```

Devteam.space

Use reviews:

- It prevents “one person projects”
- Two know more than one
- Feedback

nedap

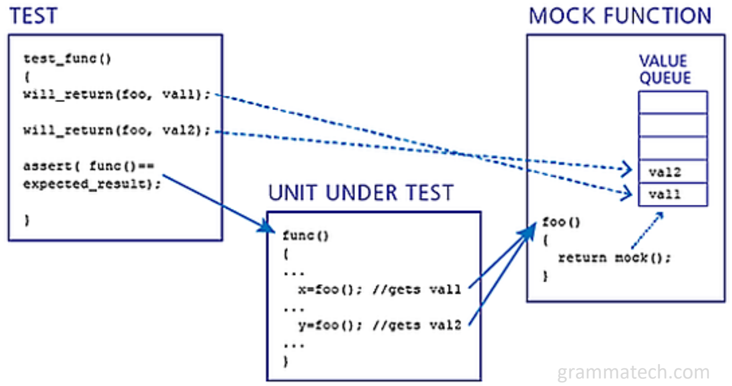
Power Electronics & Energy Storage event

POWER ELECTRONICS 2022 ENERGY STORAGE EVENT 2022

14 juni 2022 | 1931 Congrescentrum 's-Hertogenbosch

Firmware development: Code Quality

techwell.com



```
test_func()
{
will_return(foo, val1);

will_return(foo, val2);

assert( func()==
expected_result);

}
```

```
func()
{
...
  x=foo(); //gets val1
...
  y=foo(); //gets val2
...
}
```

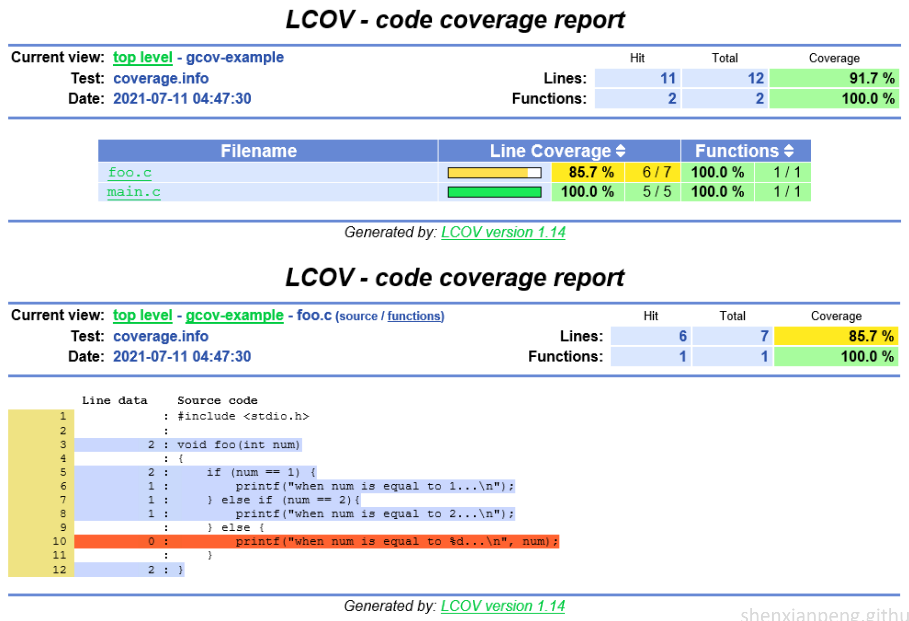
```
foo()
{
  return mock();
}
```

grammatech.com

Unit testing:

- Write unit tests for functions to test proper functionality
- Mock functions to mimic there functionality
- Automation of unit testing

Firmware development: Code Quality



LCOV - code coverage report

			Hit	Total	Coverage
Current view:	top level - gcov-example				
Test:	coverage.info	**Lines:**	11	12	91.7 %
Date:	2021-07-11 04:47:30	**Functions:**	2	2	100.0 %

Filename	Line Coverage ⬍			Functions ⬍	
foo.c		85.7 %	6 / 7	100.0 %	1 / 1
main.c		100.0 %	5 / 5	100.0 %	1 / 1

Generated by: LCOV version 1.14

LCOV - code coverage report

			Hit	Total	Coverage
Current view:	top level - gcov-example - foo.c (source / functions)				
Test:	coverage.info	**Lines:**	6	7	85.7 %
Date:	2021-07-11 04:47:30	**Functions:**	1	1	100.0 %

```
      Line data    Source code
                 : #include <stdio.h>
                 :
               2 : void foo(int num)
                 : {
               2 :     if (num == 1) {
               1 :         printf("when num is equal to 1...\n");
               1 :     } else if (num == 2){
               1 :         printf("when num is equal to 2...\n");
                 :     } else {
               0 :         printf("when num is equal to %d...\n", num);
                 :     }
               2 : }
```

Generated by: LCOV version 1.14

shenxianpeng.github.io

Code coverage:

- Generate code coverage reports to see which codelines are tested

nedap

Firmware development: Code Quality

techwell.com

noffz.com

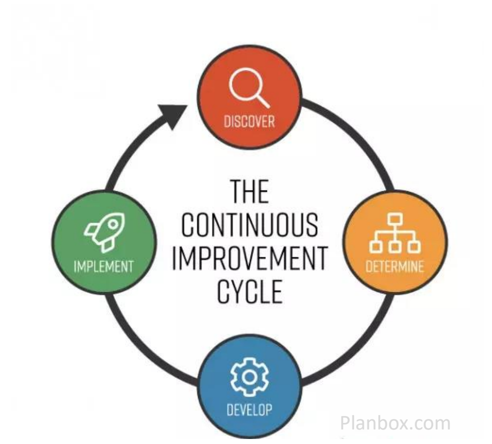
HIL tester:

- Use a Hardware in the Loop tester to automate tests
- Run tests over and over again under the same conditions

nedap

Power Electronics & Energy Storage event
POWER ELECTRONICS 2022
ENERGY STORAGE EVENT 2022
14 juni 2022 | 1931 Congrescentrum 's-Hertogenbosch

Firmware development: Code Quality



General:

- Add extra tests, while older tests still run
- Reusability of functions in other projects
- Visualize your improvements
- Keep improving the quality!!

- Nedap Light Controls info@nedap-lightcontrols.com