Inphi®

Think fast.

Inphi Moves Big Data Faster

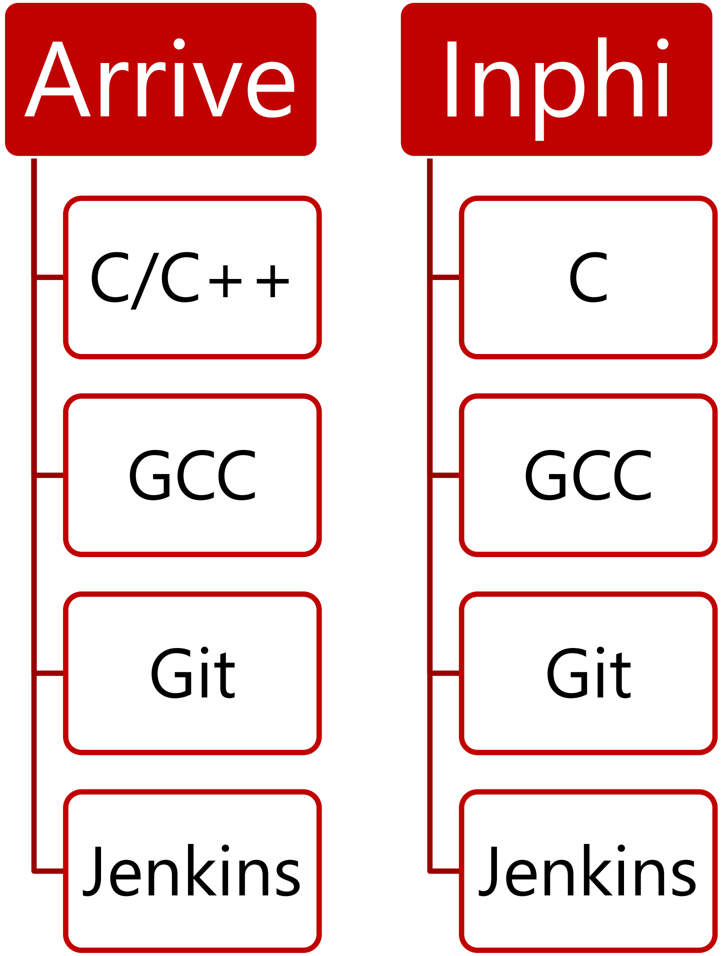
HSC SW Tools and Methodology

2020-06-01 – Devin Linnington

(Proprietary & Confidential)

Arrive and Inphi – Solving the same problems

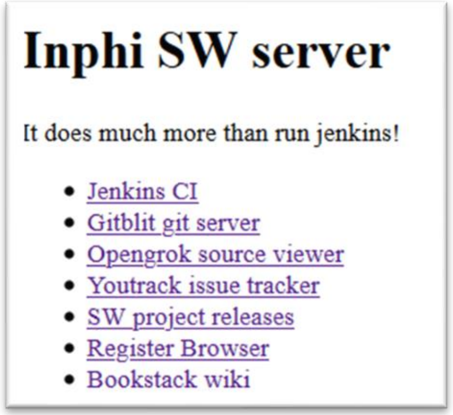
- HW-focused SW design
  - We sell HW, not SW
- Full stack
  - Embedded low-level C/C++
  - High level test frameworks and debug GUIs
- Standard SW tooling
  - Issue tracking
  - Documentation
  - Git SCM
- Good use of no-cost Open Source tooling



HSC SW Tools

SW Services

- "sw" server hosting multiple *internal* services
  - http://sw.inphi-corp.local/
  - Hosted in LAS datacenter
  - One of many HSC SW dedicated linux (RHEL) machines

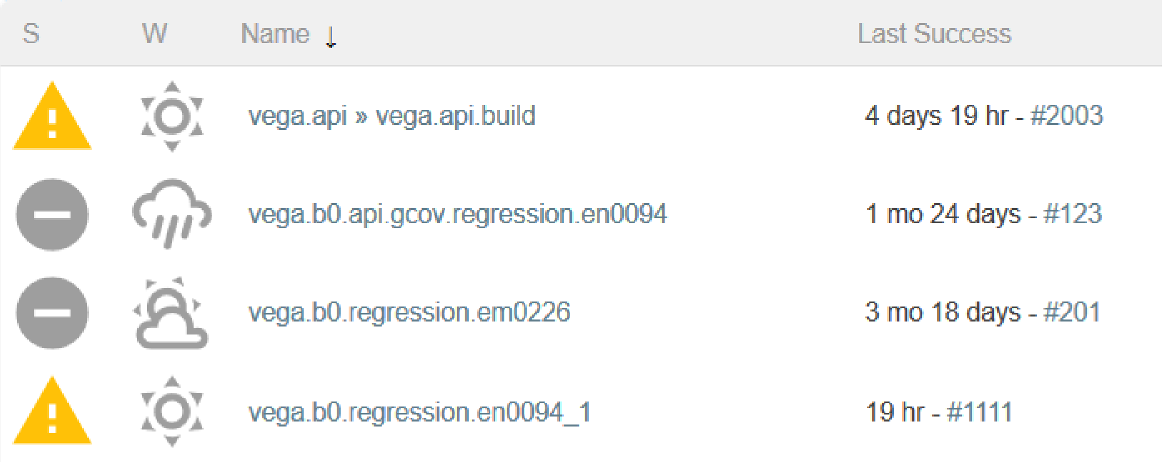


Inphi SW server

It does much more than run jenkins!

- Jenkins CI
- Gitblit git server
- Opengrok source viewer
- Youtrack issue tracker
- SW project releases
- Register Browser
- Bookstack wiki

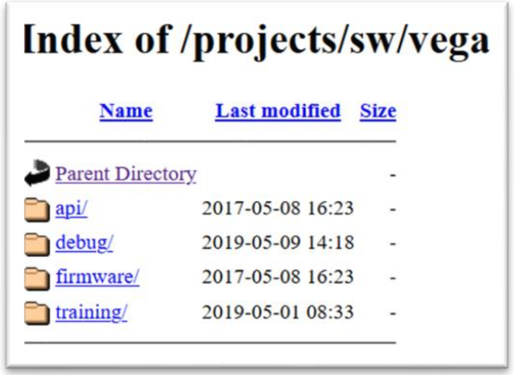
- Jenkins – "Do everything" script-runner
  - http://sw.inphi-corp.local/jenkins/
  - Post-checkin tester
  - Nightly builds for CI
  - Release package build
  - SW Test/Regression Runs
  - Infrastructure maintenance
  - Any automation task we can think of!



S	W	Name ↓	Last Success
⚠	☀	vega.api » vega.api.build	4 days 19 hr - #2003
⊖	🌧	vega.b0.api.gcov.regression.en0094	1 mo 24 days - #123
⊖	⛅	vega.b0.regression.em0226	3 mo 18 days - #201
⚠	☀	vega.b0.regression.en0094_1	19 hr - #1111

SW Services

- Gitblit & Gitlab – SCM management
  - http://sw.inphi-corp.local/gitblit/
  - http://las-gitlab.inphi-corp.local
  - Transitioning from gitblit to gitlab; old on gitblit, new on gitlab
  - Post-commit code review
  - Gitlab is common for all Inphi engineering



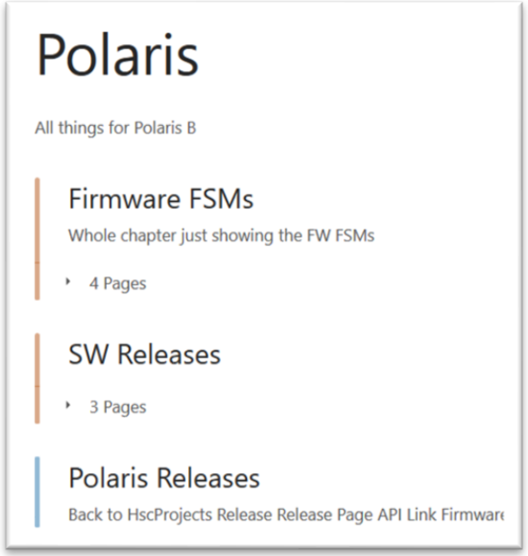
Index of /projects/sw/vega

Name	Last modified	Size
Parent Directory		-
api/	2017-05-08 16:23	-
debug/	2019-05-09 14:18	-
firmware/	2017-05-08 16:23	-
training/	2019-05-01 08:33	-

- Build Artifact Repository
  - http://sw.inphi-corp.local/projects/sw/vega/
  - Permanent storage of all releases (nightly and official)
  - Online API/SDK documentation

SW Services

- Bookstack Documentation
  - http://sw.inphi-corp.local/bookstack/
  - Wiki-replacement for documents and notes
  - WYSIWYG editor
  - Draw.io diagram support



Polaris

All things for Polaris B

Firmware FSMs

Whole chapter just showing the FW FSMs

4 Pages

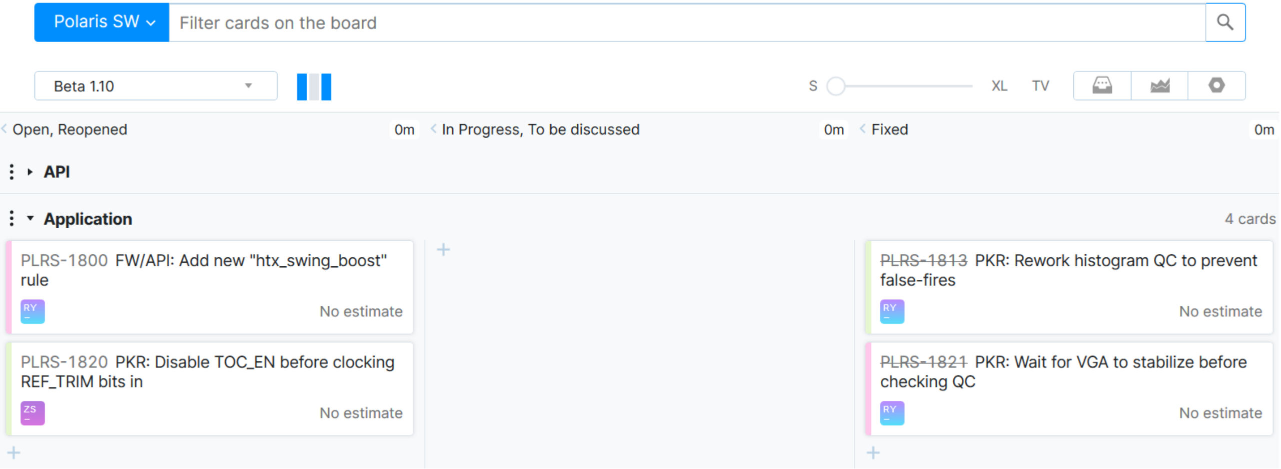
SW Releases

3 Pages

Polaris Releases

Back to HscProjects Release Release Page API Link Firmware

- Youtrack Issue Tracking
  - http://youtrack.inphi-corp.local/
  - Issue tracking, project planning, agile dashboard
  - Common for all Inphi engineering



Open, Reopened 0m | In Progress, To be discussed 0m | Fixed 0m

API

Application 4 cards

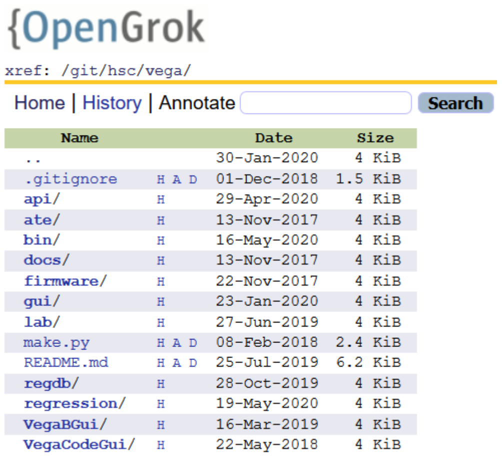
PLRS-1800 FW/API: Add new "htx_swing_boost" rule — RY — No estimate

PLRS-1820 PKR: Disable TOC_EN before clocking REF_TRIM bits in — ZS — No estimate

~~PLRS-1813~~ PKR: Rework histogram QC to prevent false-fires — RY — No estimate

~~PLRS-1821~~ PKR: Wait for VGA to stabilize before checking QC — RY — No estimate

SW Services



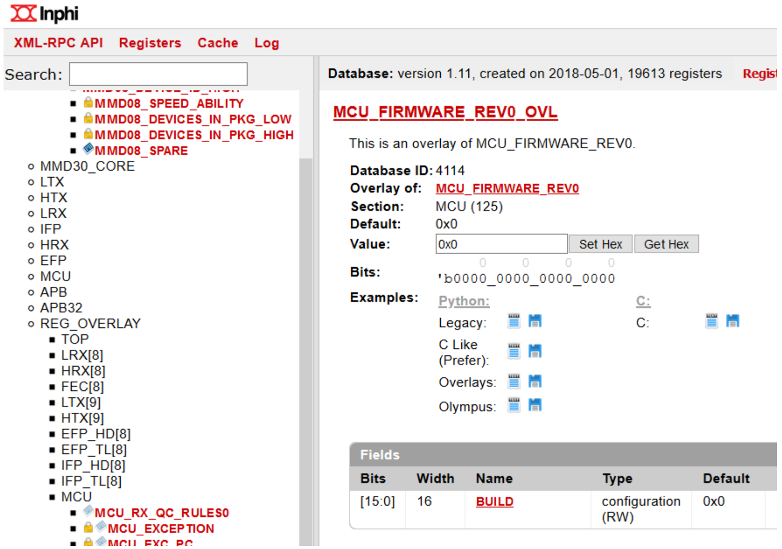
{OpenGrok

xref: /git/hsc/vega/

Home | History | Annotate [ ] Search

Name		Date	Size
..		30-Jan-2020	4 KiB
.gitignore	H A D	01-Dec-2018	1.5 KiB
api/	H	29-Apr-2020	4 KiB
ate/	H	13-Nov-2017	4 KiB
bin/	H	16-May-2020	4 KiB
docs/	H	13-Nov-2017	4 KiB
firmware/	H	22-Nov-2017	4 KiB
gui/	H	23-Jan-2020	4 KiB
lab/	H	27-Jun-2019	4 KiB
make.py	H A D	08-Feb-2018	2.4 KiB
README.md	H A D	25-Jul-2019	6.2 KiB
regdb/	H	28-Oct-2019	4 KiB
regression/	H	19-May-2020	4 KiB
VegaBGui/	H	16-Mar-2019	4 KiB
VegaCodeGui/	H	22-May-2018	4 KiB

- OpenGrok Source Browser
  - http://sw.inphi-corp.local/source/xref/git/hsc/vega/
  - For linking to and searching through project code



- MMD08_SPEED_ABILITY
- MMD08_DEVICES_IN_PKG_LOW
- MMD08_DEVICES_IN_PKG_HIGH
- MMD08_SPARE

- MMD30_CORE
- LTX
- HTX
- LRX
- IFP
- HRX
- EFP
- MCU
- APB
- APB32
- REG_OVERLAY
  - TOP
  - LRX[8]
  - HRX[8]
  - FEC[8]
  - LTX[9]
  - HTX[9]
  - EFP_HD[8]
  - EFP_TL[8]
  - IFP_HD[8]
  - IFP_TL[8]
  - MCU
    - MCU_RX_QC_RULES0
    - MCU_EXCEPTION
    - MCU_EXC_PC

Database: version 1.11, created on 2018-05-01, 19613 registers

MCU_FIRMWARE_REV0_OVL

This is an overlay of MCU_FIRMWARE_REV0.

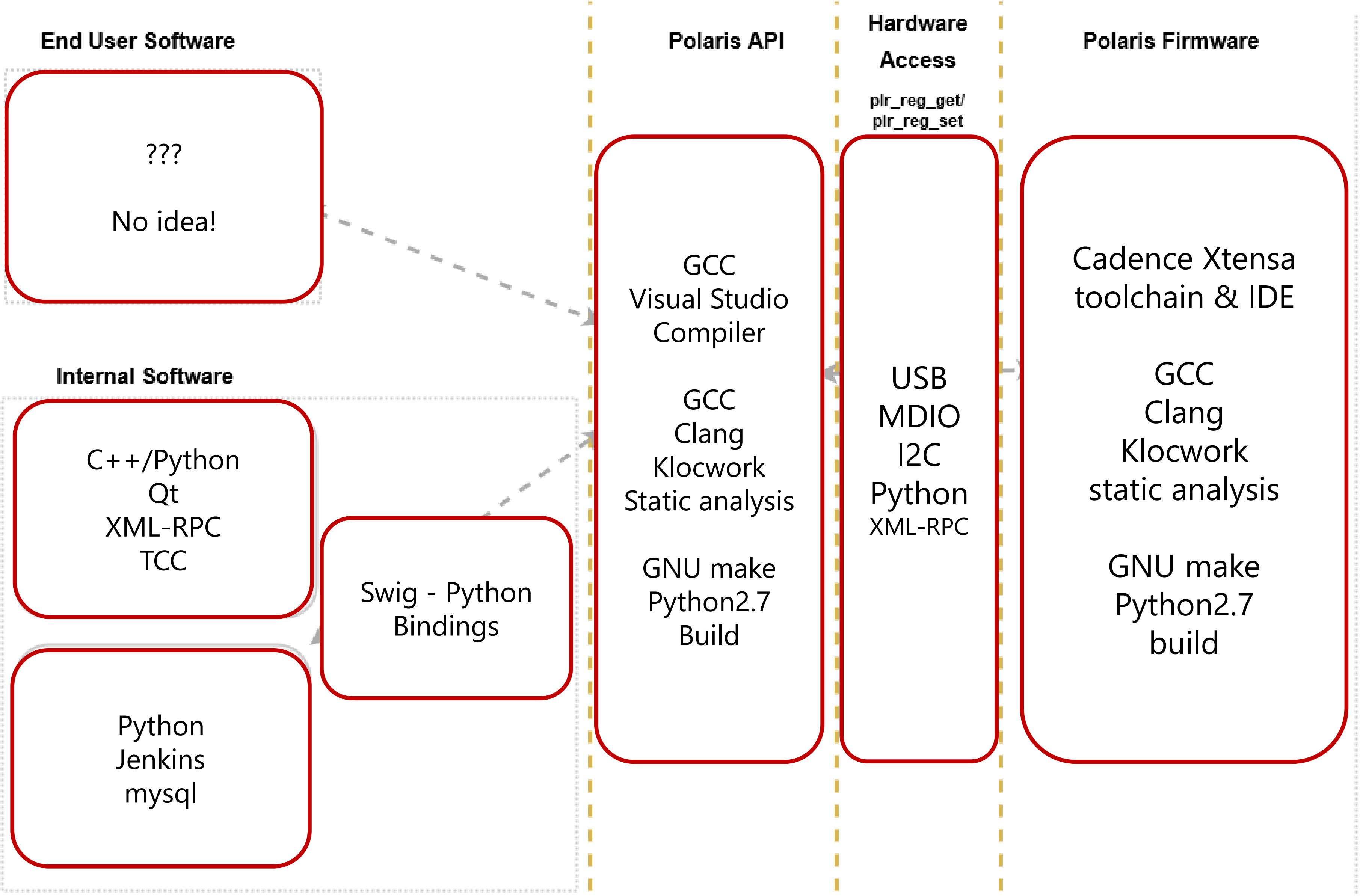
Database ID: 4114
Overlay of: MCU_FIRMWARE_REV0
Section: MCU (125)
Default: 0x0
Value: 0x0 Set Hex Get Hex
Bits: 0 0 0 0 'b0000_0000_0000_0000
Examples: Python: C:
Legacy: C:
C Like (Prefer):
Overlays:
Olympus:

Fields

Bits	Width	Name	Type	Default
[15:0]	16	BUILD	configuration (RW)	0x0

- Online Register Browser
  - http://sw.inphi-corp.local:8086/browser?chip=vegab&die=0x0
  - Derived from the actual HW spec (automated)
  - Forms basis of API/SDK/FW interaction with HW
  - Used to generate automatic HW register bindings in C/Python

Tools Overview



End User Software

???

No idea!

Internal Software

C++/Python
Qt
XML-RPC
TCC

Python
Jenkins
mysql

Swig - Python Bindings

Polaris API

GCC
Visual Studio Compiler

GCC
Clang
Klocwork
Static analysis

GNU make
Python2.7
Build

Hardware Access

plr_reg_get/
plr_reg_set

USB
MDIO
I2C
Python
XML-RPC

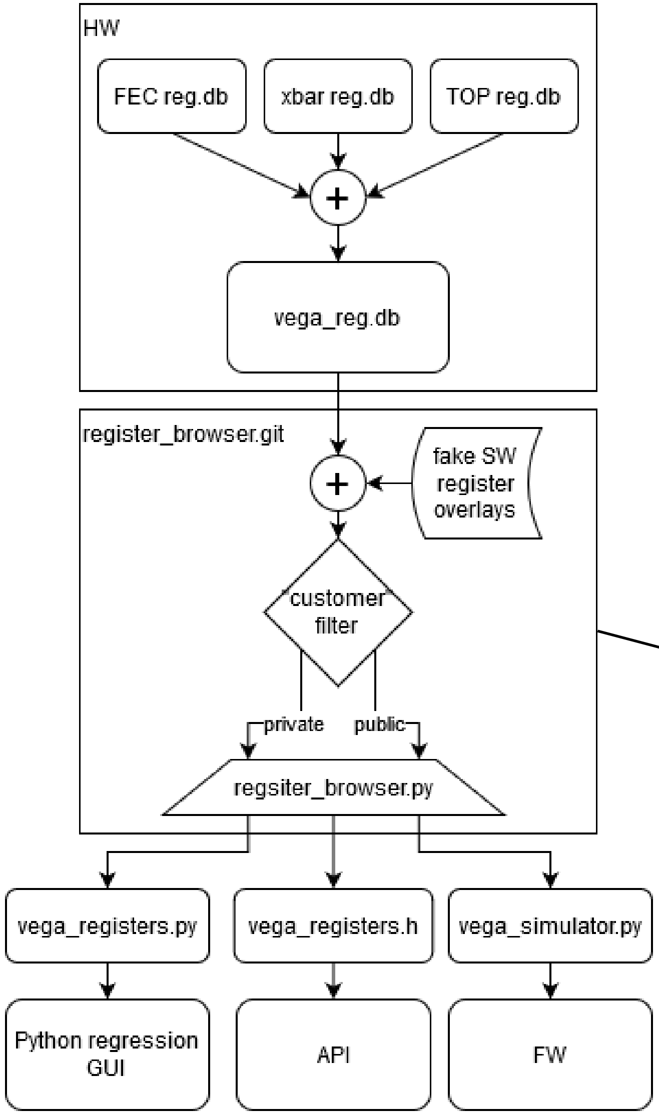
Polaris Firmware

Cadence Xtensa toolchain & IDE

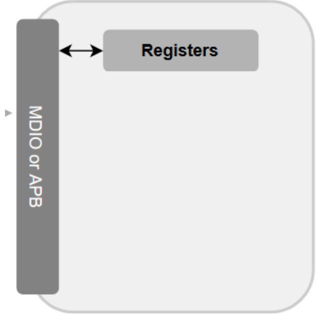
GCC
Clang
Klocwork
static analysis

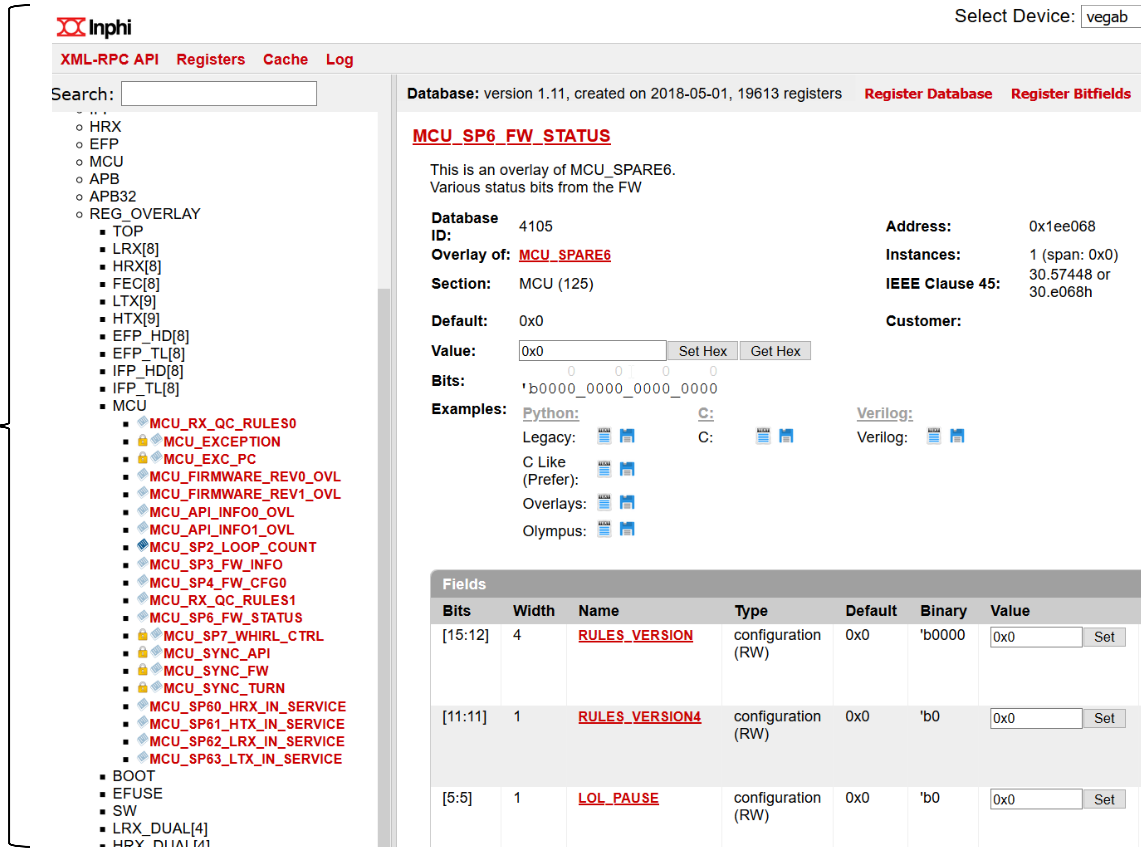
GNU make
Python2.7
build

Reg.db Register Database



- HW designers build a "reg.db" sqlite database
- SW team adds overlays and filters registers based on public/private info
- Internal Source Link





Database: version 1.11, created on 2018-05-01, 19613 registers

MCU_SP6_FW_STATUS

This is an overlay of MCU_SPARE6.
Various status bits from the FW

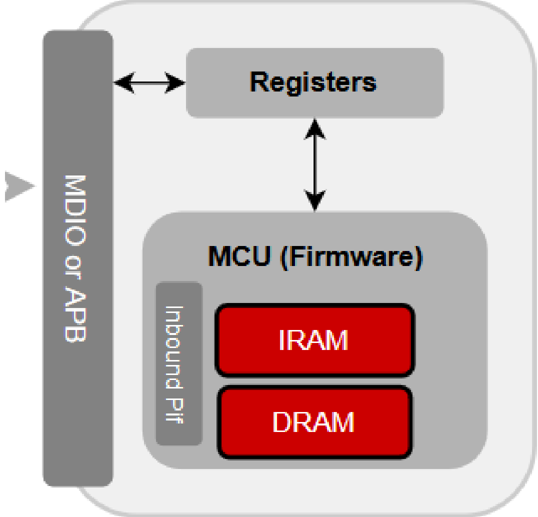
Database ID:	4105	**Address:**	0x1ee068
Overlay of:	MCU_SPARE6	**Instances:**	1 (span: 0x0)
Section:	MCU (125)	**IEEE Clause 45:**	30.57448 or 30.e068h
Default:	0x0	**Customer:**	
Value:	0x0		
Bits:	'b0000_0000_0000_0000		

Examples: Python: Legacy:, C Like (Prefer):, Overlays:, Olympus:; C: C:; Verilog: Verilog:

Fields

Bits	Width	Name	Type	Default	Binary	Value
[15:12]	4	RULES_VERSION	configuration (RW)	0x0	'b0000	0x0 Set
[11:11]	1	RULES_VERSION4	configuration (RW)	0x0	'b0	0x0 Set
[5:5]	1	LOL_PAUSE	configuration (RW)	0x0	'b0	0x0 Set

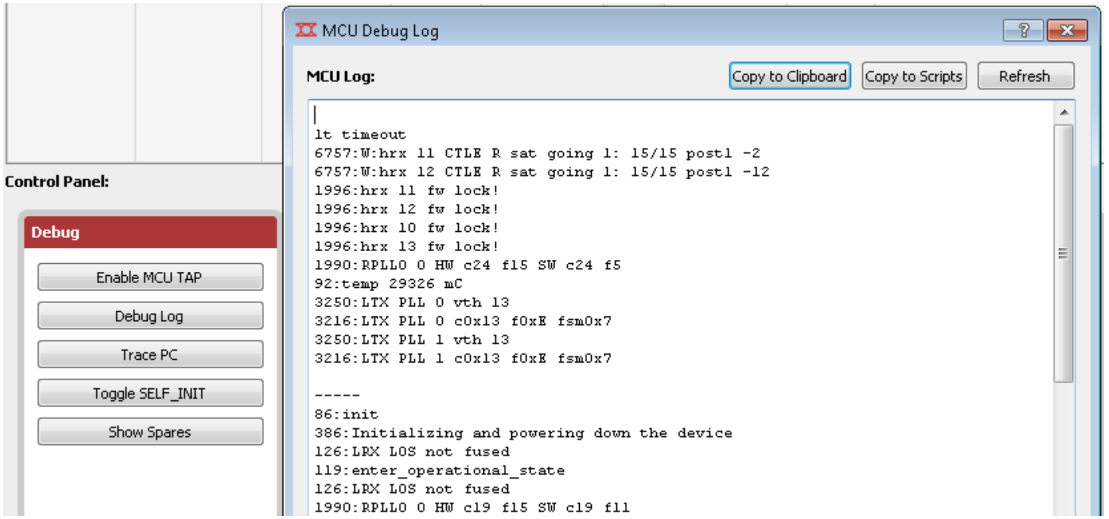
FW Tools



- C99 with a few GNU extensions (limited use)
- HW MCU Compiler: Cadence Xtensa SDK & IDE
- Simulation: Visual Studio or GCC
  - Any C/C++ compiler should work
- Build: custom GNU makefiles and Python helper scripts
- Static Analysis: GCC, clang (scan-build), Perforce Klocwork

- Custom debug tools:
  - MCU/FW print log
  - libwhirl data logging
  - HW debug GUI
  - Many, many more!

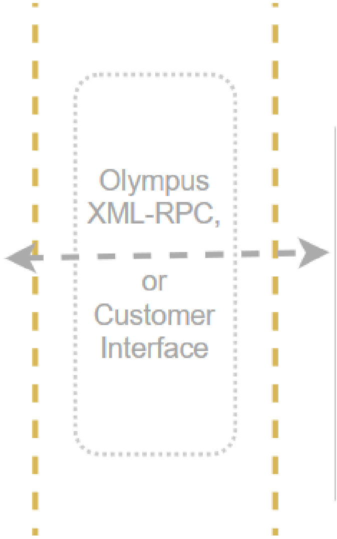
- Internal Source Link



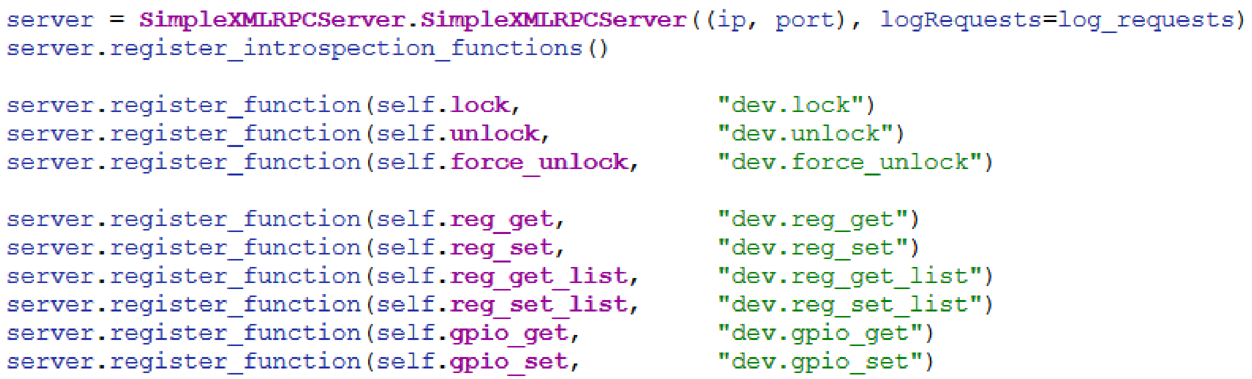
```
lt timeout
6757:W:hrx 11 CTLE R sat going 1: 15/15 post1 -2
6757:W:hrx 12 CTLE R sat going 1: 15/15 post1 -12
1996:hrx 11 fw lock!
1996:hrx 12 fw lock!
1996:hrx 10 fw lock!
1996:hrx 13 fw lock!
1990:RPLL0 0 HW c24 f15 SW c24 f5
92:temp 29326 mC
3250:LTX PLL 0 vth 13
3216:LTX PLL 0 c0x13 f0xE fsm0x7
3250:LTX PLL 1 vth 13
3216:LTX PLL 1 c0x13 f0xE fsm0x7

-----
86:init
386:Initializing and powering down the device
126:LRX LOS not fused
119:enter_operational_state
126:LRX LOS not fused
1990:RPLL0 0 HW c19 f15 SW c19 f11
```

HW Access

- Primary access to HW config/state is done via registers
  - Some interrupt pins and GPIOs, but mostly registers
- Real HW connects externally via I2C or MDIO (selectable)
- Eval board FPGA converts I2C/MDIO to USB with custom protocol (Olympus)
- XML-RPC server/client implements a network wrapper around Olympus
  - Used internally for testing, connecting to multiple HW setups in a single test
  - Customers implement this wrapper too!

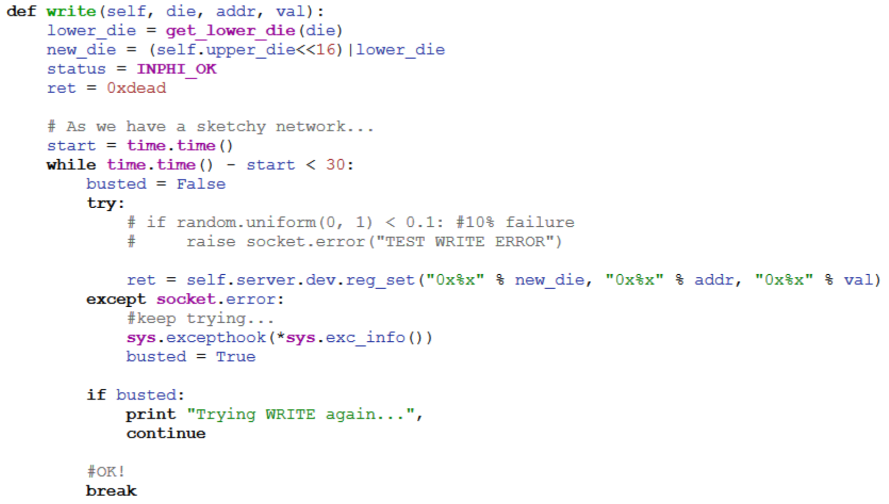


Server (link to src)



```
server = SimpleXMLRPCServer.SimpleXMLRPCServer((ip, port), logRequests=log_requests)
server.register_introspection_functions()

server.register_function(self.lock,               "dev.lock")
server.register_function(self.unlock,             "dev.unlock")
server.register_function(self.force_unlock,       "dev.force_unlock")

server.register_function(self.reg_get,            "dev.reg_get")
server.register_function(self.reg_set,            "dev.reg_set")
server.register_function(self.reg_get_list,       "dev.reg_get_list")
server.register_function(self.reg_set_list,       "dev.reg_set_list")
server.register_function(self.gpio_get,           "dev.gpio_get")
server.register_function(self.gpio_set,           "dev.gpio_set")
```

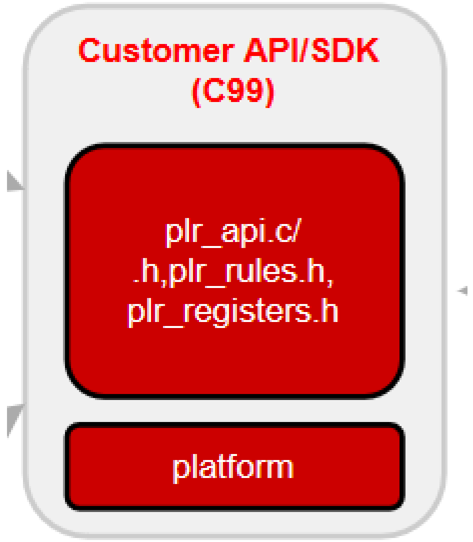
Client (link to src)



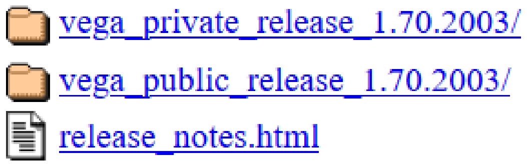
```
def write(self, die, addr, val):
    lower_die = get_lower_die(die)
    new_die = (self.upper_die<<16)|lower_die
    status = INPHI_OK
    ret = 0xdead

    # As we have a sketchy network...
    start = time.time()
    while time.time() - start < 30:
        busted = False
        try:
            # if random.uniform(0, 1) < 0.1: #10% failure
            #     raise socket.error("TEST WRITE ERROR")

            ret = self.server.dev.reg_set("0x%x" % new_die, "0x%x" % addr, "0x%x" % val)
        except socket.error:
            #keep trying...
            sys.excepthook(*sys.exc_info())
            busted = True

        if busted:
            print "Trying WRITE again...",
            continue

        #OK!
        break
```

API/SDK

- Commonly referred to as the “API”
- Given to customer as ISO C99 source code
- Any C compiler, any environment, *has to work*
  - “platform” dir provides abstraction around stdlib, RTOS
- Build: custom GNU makefiles and Python helper scripts
- Static Analysis: GCC, clang (scan-build), Perforce Klocwork



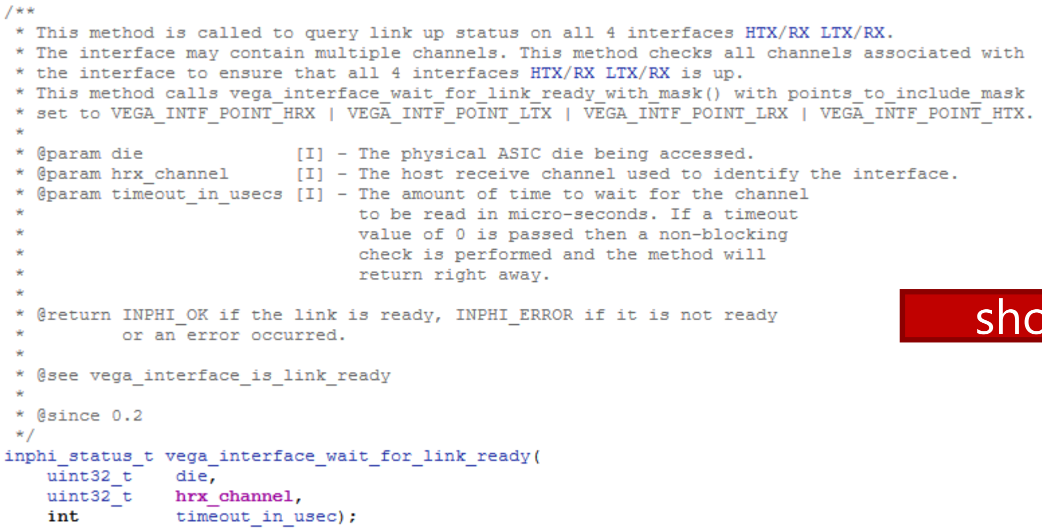
- Lots of custom packaging for customer builds
  - Concept of public (customer) and private (internal) builds
  - Registers, code, documentation is all filtered around this


vega_private_release_1.70.2003/
vega_public_release_1.70.2003/
release_notes.html

- Internal Source Link

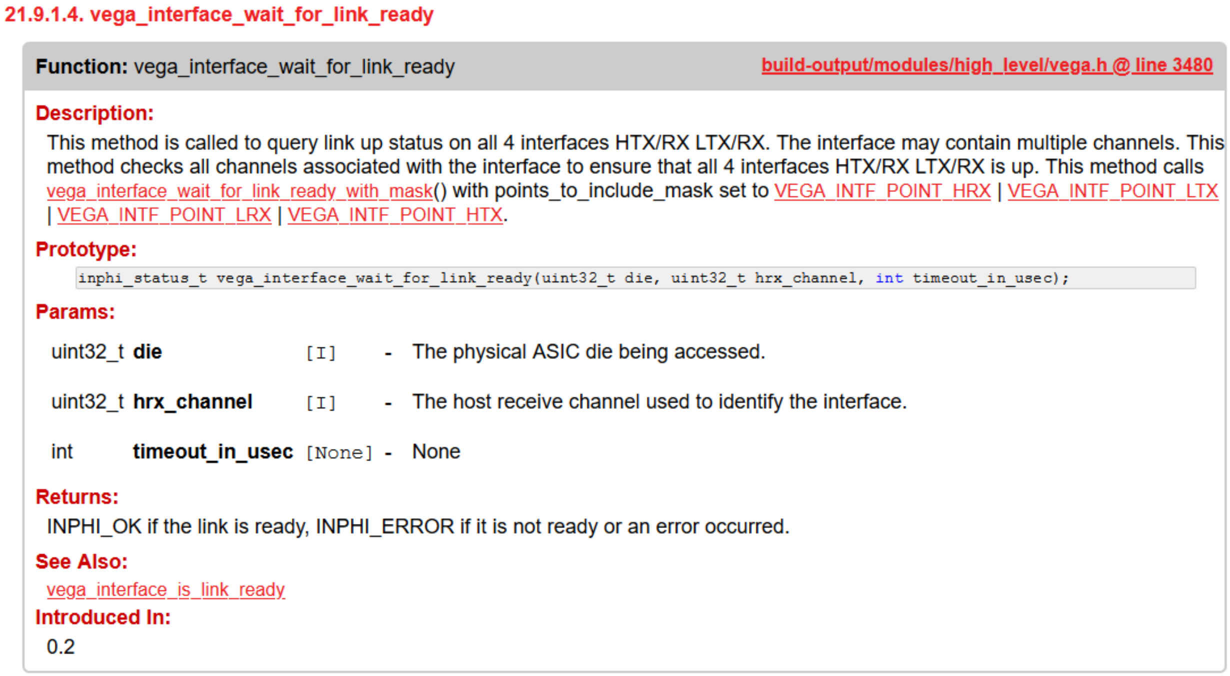
API/SDK shorte Documentation

- Custom markup language called shorte
- shorte parser written in python
- Pulls in C doc-comments, user written chapters, release notes, and generates HTML/PDF documentation for internal/external users



```
/**
 * This method is called to query link up status on all 4 interfaces HTX/RX LTX/RX.
 * The interface may contain multiple channels. This method checks all channels associated with
 * the interface to ensure that all 4 interfaces HTX/RX LTX/RX is up.
 * This method calls vega_interface_wait_for_link_ready_with_mask() with points_to_include_mask
 * set to VEGA_INTF_POINT_HRX | VEGA_INTF_POINT_LTX | VEGA_INTF_POINT_LRX | VEGA_INTF_POINT_HTX.
 *
 * @param die              [I] - The physical ASIC die being accessed.
 * @param hrx_channel      [I] - The host receive channel used to identify the interface.
 * @param timeout_in_usecs [I] - The amount of time to wait for the channel
 *                               to be read in micro-seconds. If a timeout
 *                               value of 0 is passed then a non-blocking
 *                               check is performed and the method will
 *                               return right away.
 *
 * @return INPHI_OK if the link is ready, INPHI_ERROR if it is not ready
 *         or an error occurred.
 *
 * @see vega_interface_is_link_ready
 *
 * @since 0.2
 */
inphi_status_t vega_interface_wait_for_link_ready(
    uint32_t    die,
    uint32_t    hrx_channel,
    int         timeout_in_usec);
```

shorte



21.9.1.4. vega_interface_wait_for_link_ready

Function: vega_interface_wait_for_link_ready build-output/modules/high_level/vega.h @ line 3480

Description:

This method is called to query link up status on all 4 interfaces HTX/RX LTX/RX. The interface may contain multiple channels. This method checks all channels associated with the interface to ensure that all 4 interfaces HTX/RX LTX/RX is up. This method calls vega_interface_wait_for_link_ready_with_mask() with points_to_include_mask set to VEGA_INTF_POINT_HRX | VEGA_INTF_POINT_LTX | VEGA_INTF_POINT_LRX | VEGA_INTF_POINT_HTX.

Prototype:

```
inphi_status_t vega_interface_wait_for_link_ready(uint32_t die, uint32_t hrx_channel, int timeout_in_usec);
```

Params:

uint32_t	**die**	[I]	-	The physical ASIC die being accessed.
uint32_t	**hrx_channel**	[I]	-	The host receive channel used to identify the interface.
int	**timeout_in_usec**	[None]	-	None

Returns:

INPHI_OK if the link is ready, INPHI_ERROR if it is not ready or an error occurred.

See Also:

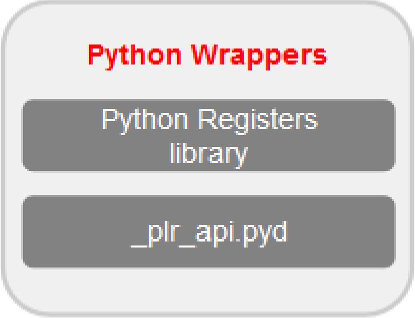
vega_interface_is_link_ready

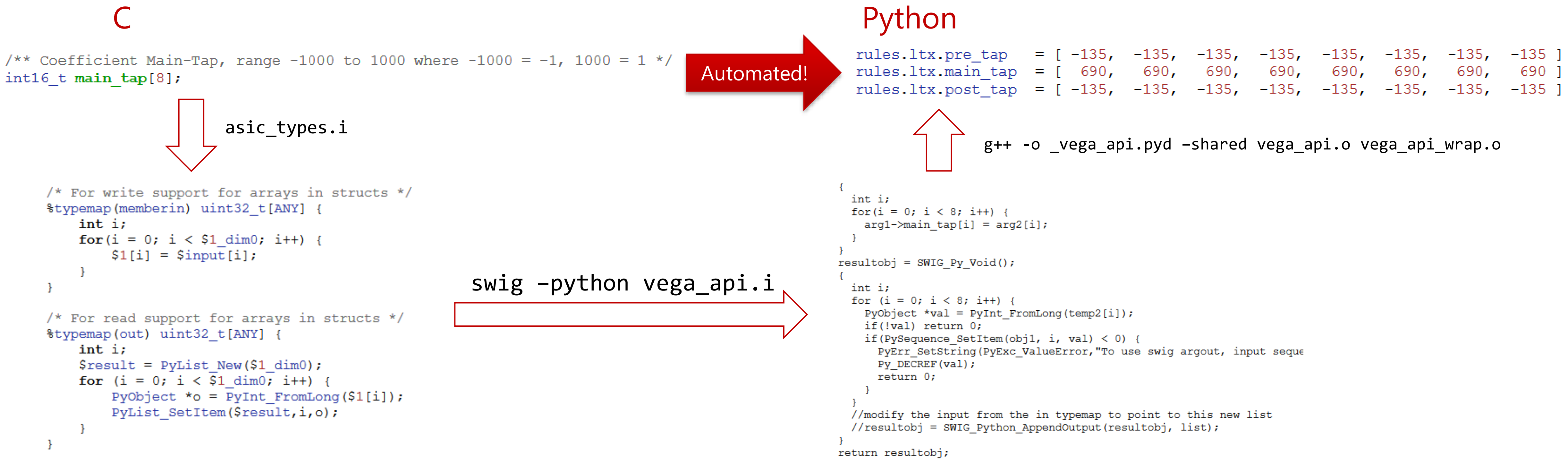
Introduced In:

0.2

Swig Python Bindings

- Awful to test in plain C; spice it up with Python!
- 100’s of functions in the API, create automatic bindings with swig
- Handful of custom functions for hard-to-translate functionality





```c
/** Coefficient Main-Tap, range -1000 to 1000 where -1000 = -1, 1000 = 1 */
int16_t main_tap[8];
```

```c
/* For write support for arrays in structs */
%typemap(memberin) uint32_t[ANY] {
    int i;
    for(i = 0; i < $1_dim0; i++) {
        $1[i] = $input[i];
    }
}

/* For read support for arrays in structs */
%typemap(out) uint32_t[ANY] {
    int i;
    $result = PyList_New($1_dim0);
    for (i = 0; i < $1_dim0; i++) {
        PyObject *o = PyInt_FromLong($1[i]);
        PyList_SetItem($result,i,o);
    }
}
```

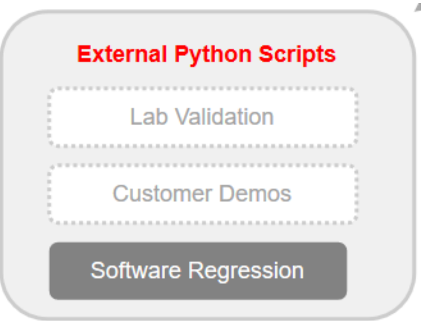
```
swig -python vega_api.i
```

```c
{
  int i;
  for(i = 0; i < 8; i++) {
    arg1->main_tap[i] = arg2[i];
  }
}
resultobj = SWIG_Py_Void();
{
  int i;
  for (i = 0; i < 8; i++) {
    PyObject *val = PyInt_FromLong(temp2[i]);
    if(!val) return 0;
    if(PySequence_SetItem(obj1, i, val) < 0) {
      PyErr_SetString(PyExc_ValueError,"To use swig argout, input seque
      Py_DECREF(val);
      return 0;
    }
  }
  //modify the input from the in typemap to point to this new list
  //resultobj = SWIG_Python_AppendOutput(resultobj, list);
}
return resultobj;
```

```
g++ -o _vega_api.pyd -shared vega_api.o vega_api_wrap.o
```

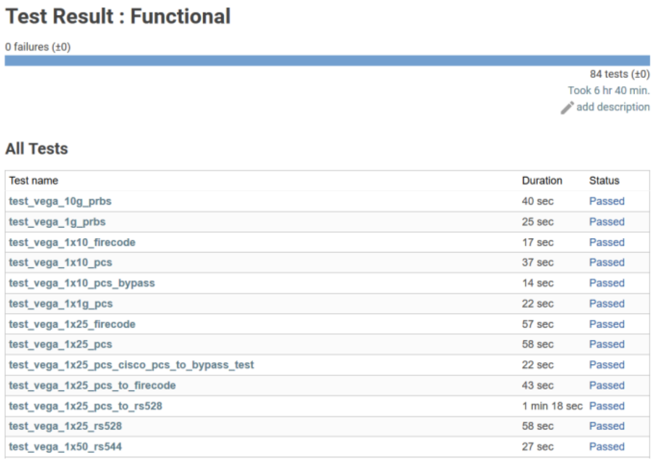
```
rules.ltx.pre_tap   = [ -135,  -135,  -135,  -135,  -135,  -135,  -135,  -135 ]
rules.ltx.main_tap  = [  690,   690,   690,   690,   690,   690,   690,   690 ]
rules.ltx.post_tap  = [ -135,  -135,  -135,  -135,  -135,  -135,  -135,  -135 ]
```

Regression & Lab Validation Tools



- SW test done in a custom "regression" environment built on Python
- Utilizes swig API wrappers, XML-RPC/Olympus, Jenkins

- Lots of cool custom utilities
  - Test equipment control
    - Temperature, voltage, ONT, etc.
  - Database data logging
    - mysql, >300k rows per product
  - Data analysis
  - Junit test reporting



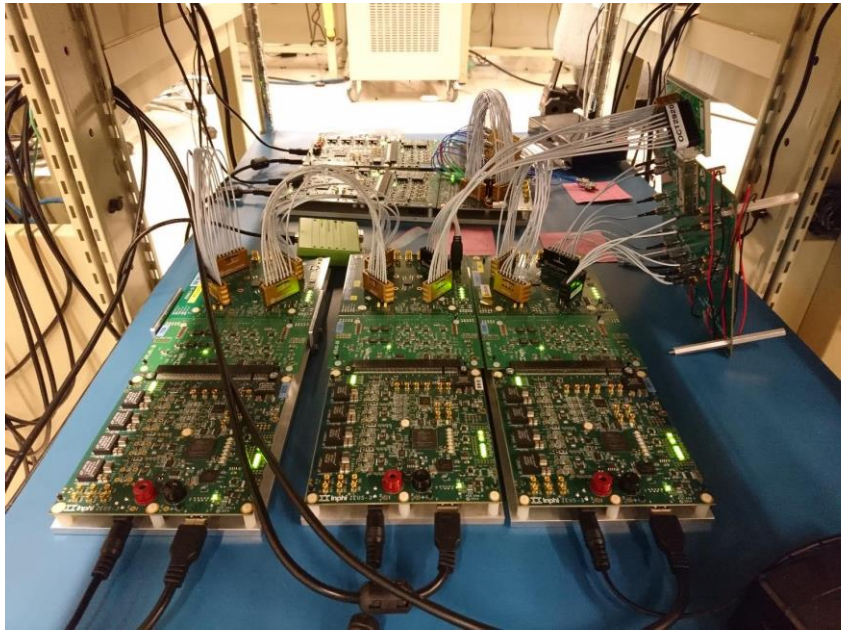
Test Result : Functional

0 failures (±0)

84 tests (±0)
Took 6 hr 40 min.
add description

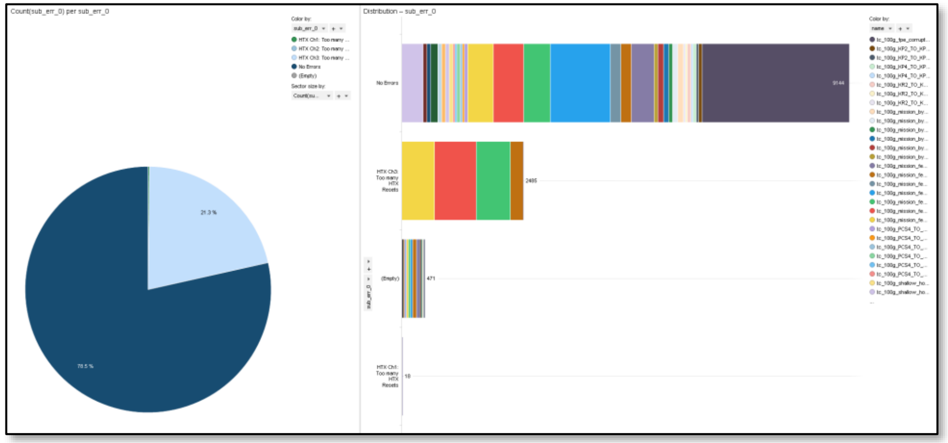
All Tests

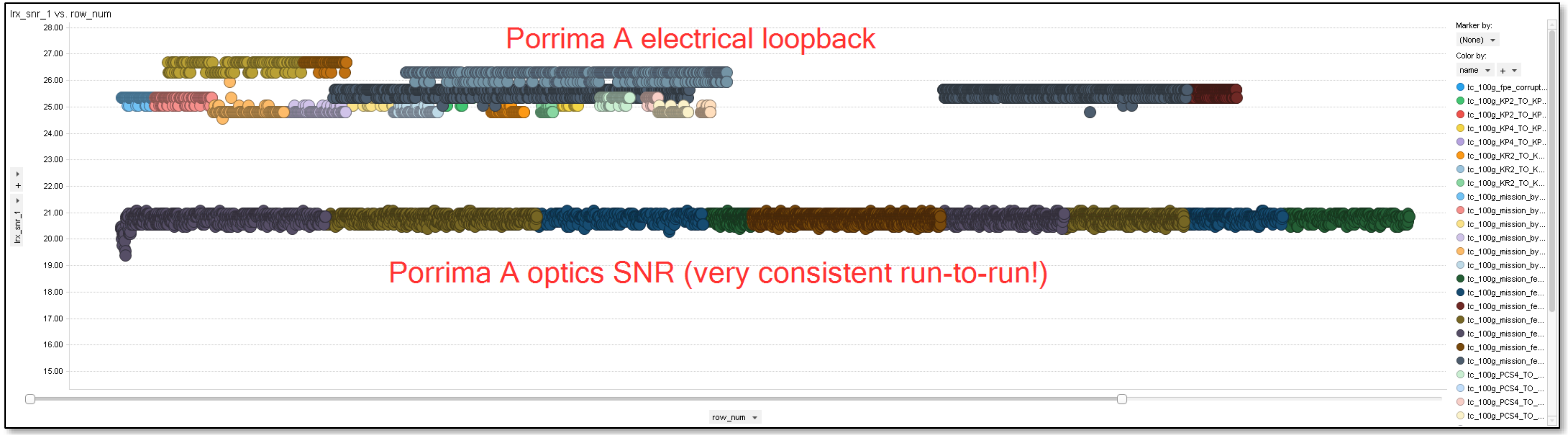
Test name	Duration	Status
test_vega_10g_prbs	40 sec	Passed
test_vega_1g_prbs	25 sec	Passed
test_vega_1x10_firecode	17 sec	Passed
test_vega_1x10_pcs	37 sec	Passed
test_vega_1x10_pcs_bypass	14 sec	Passed
test_vega_1x1g_pcs	22 sec	Passed
test_vega_1x25_firecode	57 sec	Passed
test_vega_1x25_pcs	58 sec	Passed
test_vega_1x25_pcs_cisco_pcs_to_bypass_test	22 sec	Passed
test_vega_1x25_pcs_to_firecode	43 sec	Passed
test_vega_1x25_pcs_to_rs528	1 min 18 sec	Passed
test_vega_1x25_rs528	58 sec	Passed
test_vega_1x50_rs544	27 sec	Passed



Test Results Database

- Records ~300 Test Params / Stats for Each Iteration
  - SNR, BER, VGA, time to lock, etc.
- 10-30k Stress Cases per Nightly Regression Run
- Tibco Spotfire for Visualization & Rapid Debug of Rare Events





Porrima A electrical loopback

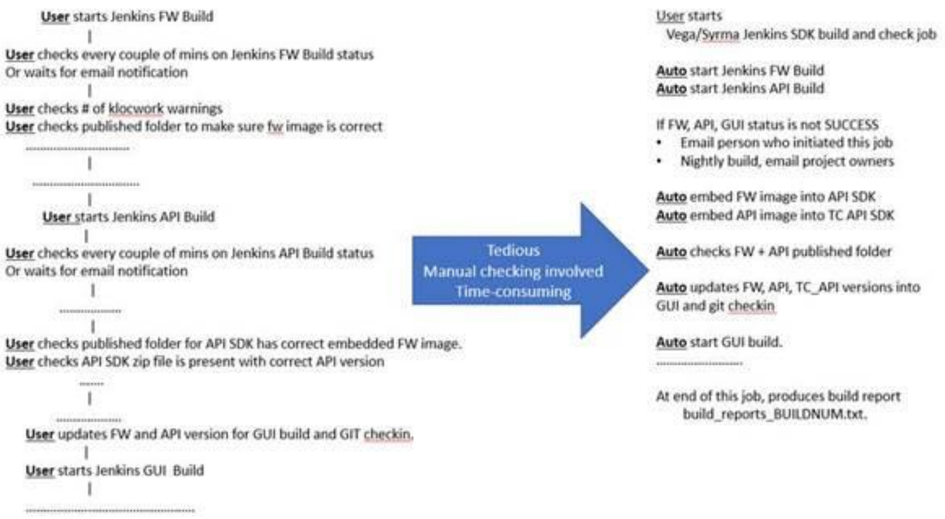
Porrima A optics SNR (very consistent run-to-run!)

Build Infrastructure

- Hand written GNU makefiles ☹
- Lots of python scripts to do complicated search & replace, filtering, etc.
- All runs in/through Jenkins

- Ongoing effort to tighten/improve Jenkins build flow

Unified release script



User starts Jenkins FW Build
|
User checks every couple of mins on Jenkins FW Build status
Or waits for email notification
|
User checks # of klocwork warnings
User checks published folder to make sure fw image is correct
....
|
....
|
User starts Jenkins API Build
|
User checks every couple of mins on Jenkins API Build status
Or waits for email notification
|
....
|
User checks published folder for API SDK has correct embedded FW image.
User checks API SDK zip file is present with correct API version
....
|
....
User updates FW and API version for GUI build and GIT checkin.
|
User starts Jenkins GUI Build
|
....

User starts
Vega/Syrma Jenkins SDK build and check job

Auto start Jenkins FW Build
Auto start Jenkins API Build

If FW, API, GUI status is not SUCCESS
- Email person who initiated this job
- Nightly build, email project owners

Auto embed FW image into API SDK
Auto embed API image into TC API SDK

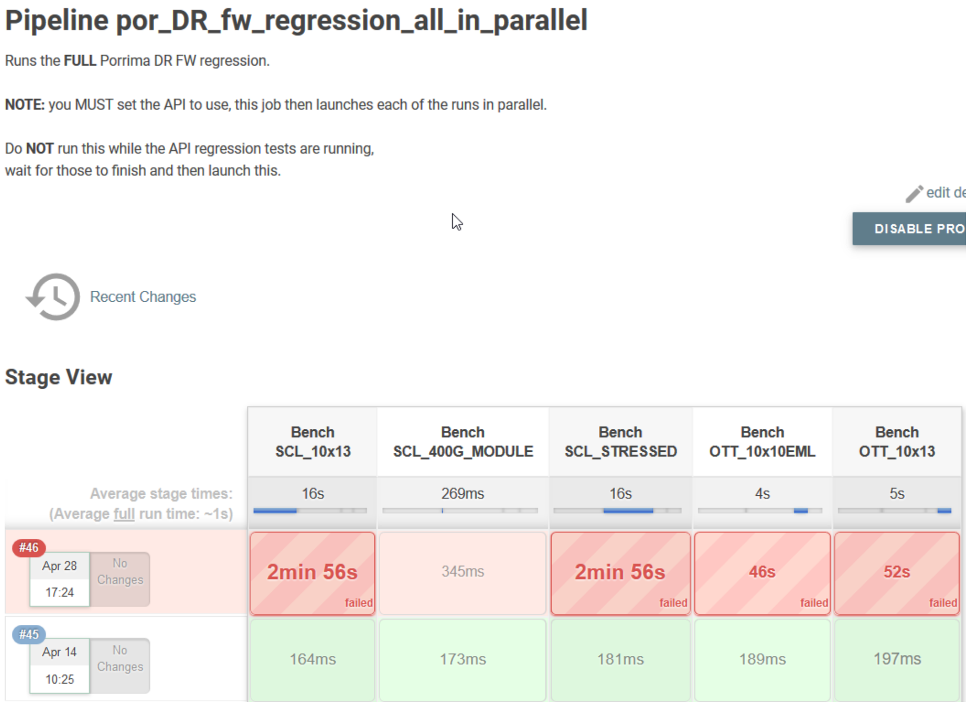
Auto checks FW + API published folder

Auto updates FW, API, TC_API versions into GUI and git checkin

Auto start GUI build.
....

At end of this job, produces build report
build_reports_BUILDNUM.txt.

Pipeline jobs



Pipeline por_DR_fw_regression_all_in_parallel

Runs the **FULL** Porrima DR FW regression.

NOTE: you MUST set the API to use, this job then launches each of the runs in parallel.

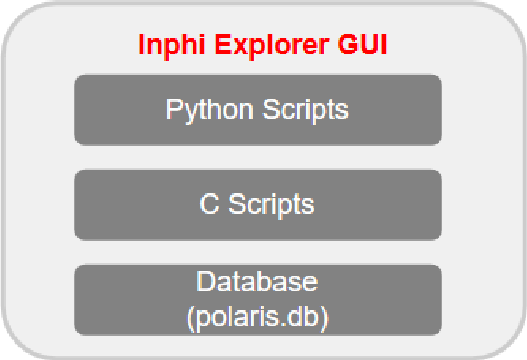
Do **NOT** run this while the API regression tests are running,
wait for those to finish and then launch this.

Stage View

	Bench SCL_10x13	Bench SCL_400G_MODULE	Bench SCL_STRESSED	Bench OTT_10x10EML	Bench OTT_10x13
Average stage times: (Average full run time: ~1s)	16s	269ms	16s	4s	5s
#46 Apr 28 17:24 No Changes	2min 56s failed	345ms	2min 56s failed	46s failed	52s failed
#45 Apr 14 10:25 No Changes	164ms	173ms	181ms	189ms	197ms

Inphi Explorer GUI

- See Brad Elliot’s HSC API & GUI presentation!

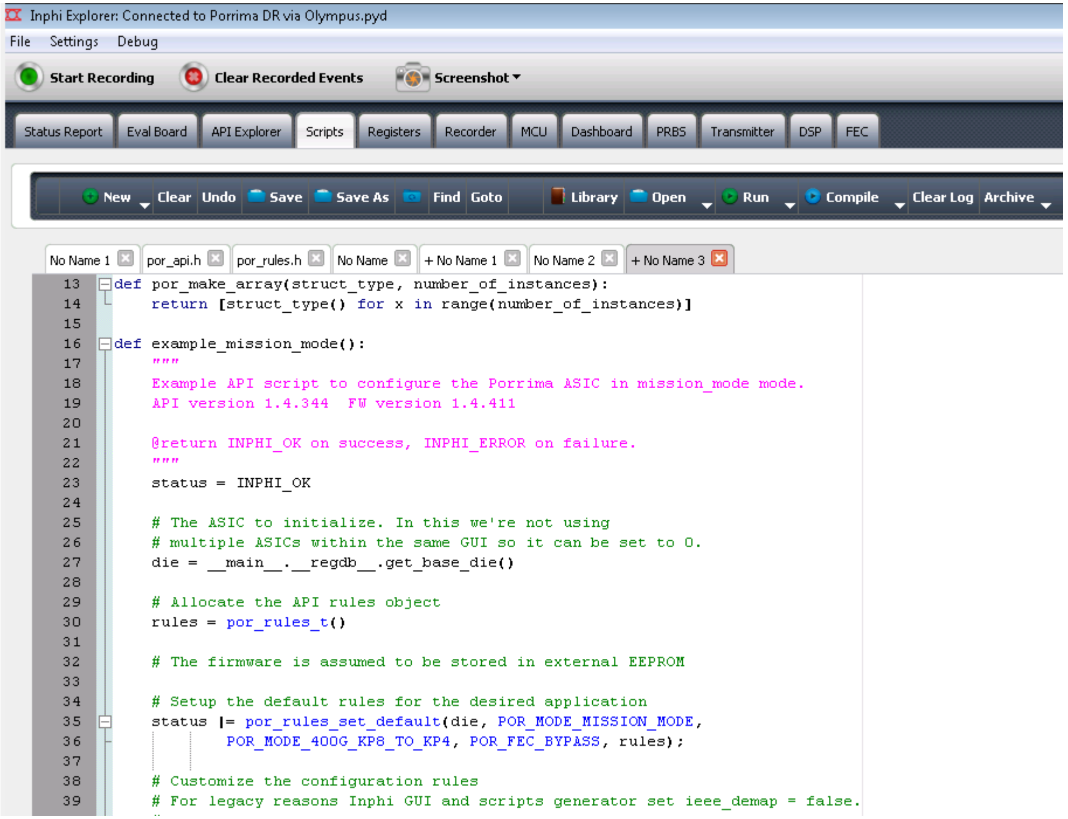


Inphi Explorer GUI

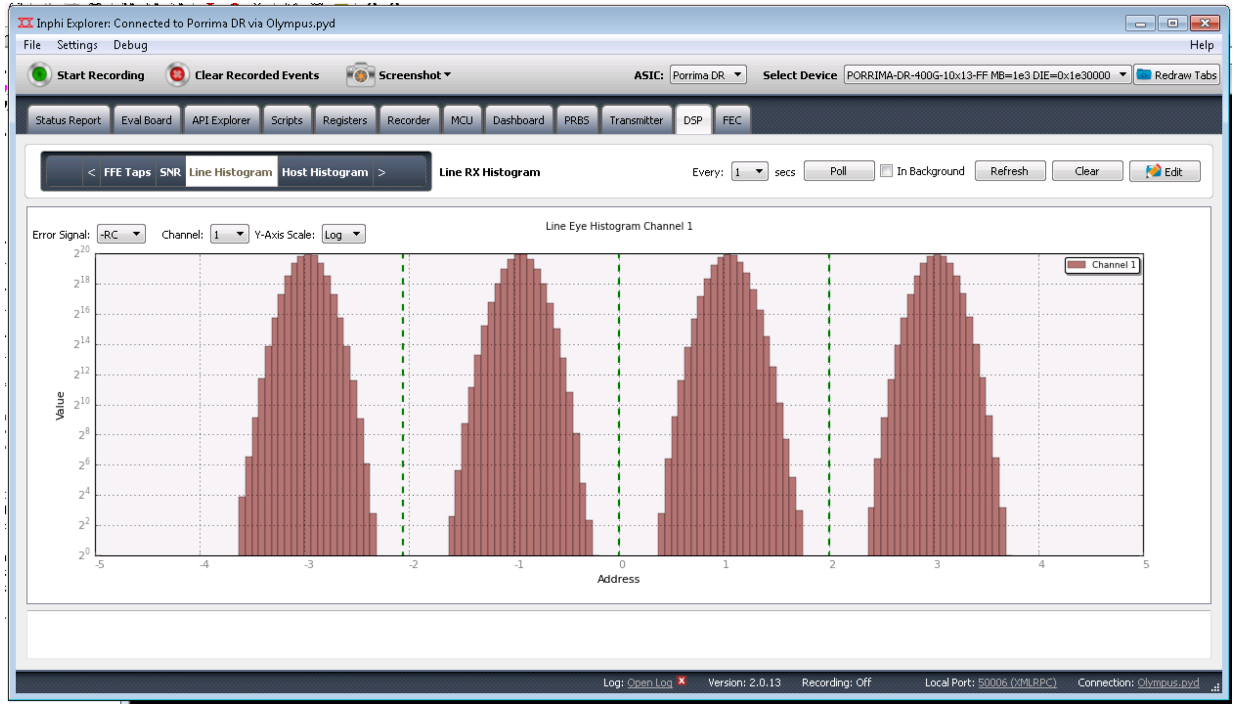
Python Scripts

C Scripts

Database (polaris.db)



```python
def por_make_array(struct_type, number_of_instances):
    return [struct_type() for x in range(number_of_instances)]

def example_mission_mode():
    """
    Example API script to configure the Porrima ASIC in mission_mode mode.
    API version 1.4.344  FW version 1.4.411

    @return INPHI_OK on success, INPHI_ERROR on failure.
    """
    status = INPHI_OK

    # The ASIC to initialize. In this we're not using
    # multiple ASICs within the same GUI so it can be set to 0.
    die = __main__.__regdb__.get_base_die()

    # Allocate the API rules object
    rules = por_rules_t()

    # The firmware is assumed to be stored in external EEPROM

    # Setup the default rules for the desired application
    status |= por_rules_set_default(die, POR_MODE_MISSION_MODE,
            POR_MODE_400G_KP8_TO_KP4, POR_FEC_BYPASS, rules);

    # Customize the configuration rules
    # For legacy reasons Inphi GUI and scripts generator set ieee_demap = false.
```



ASIC: Porrima DR

Select Device: PORRIMA-DR-400G-10x13-FF MB=1e3 DIE=0x1e30000

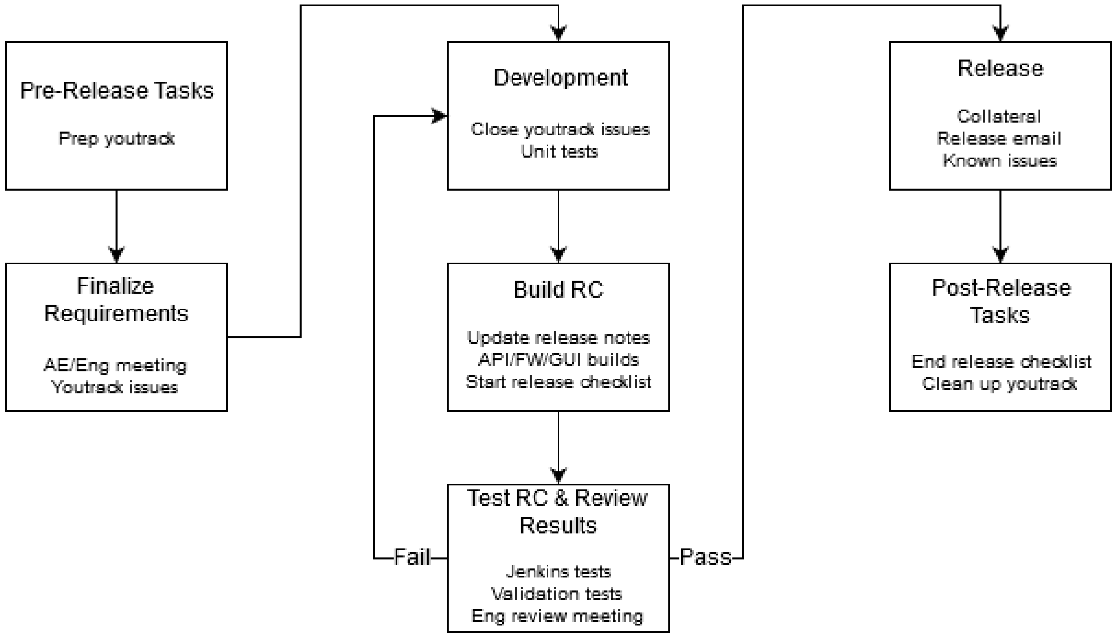
Every: 1 secs

Error Signal: -RC Channel: 1 Y-Axis Scale: Log

Log: Open Log Version: 2.0.13 Recording: Off Local Port: 50006 (XMLRPC) Connection: Olympus.pyd

HSC SW Methodology

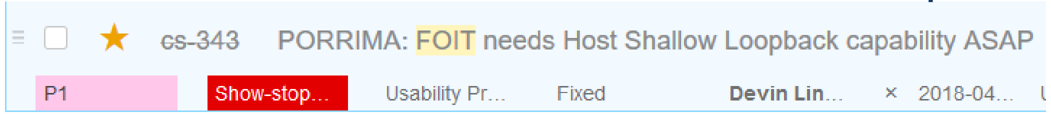
SW Development Flow



- Release flow built around issue tracking and regression testing
- All development tracked through git and Youtrack issues
- Each phase is done in coordination with other teams (AEs, Validation, HW designers, etc.)
- Testing phase includes SW regression (repeatability) and HW validation (performance)
- Release listed here is *internal;* AEs do the final release to customers (web portal, release notes, etc.)

Customer Issue/Change Management

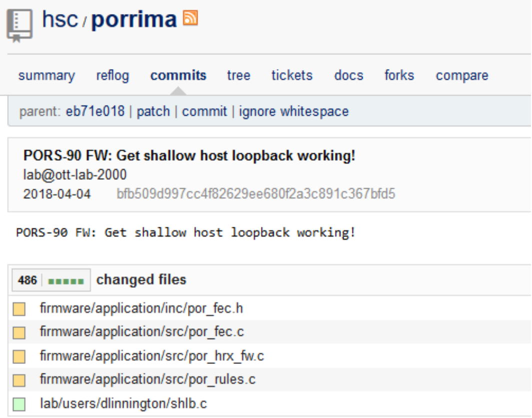
#1: AEs make new Youtrack issue/feature request



≡ □ ★ ~~cs-343~~ PORRIMA: FOIT needs Host Shallow Loopback capability ASAP

P1	Show-stop…	Usability Pr…	Fixed	Devin Lin…	×	2018-04…

#2: Development of fix/feature in git



hsc / **porrima**

parent: eb71e018 | patch | commit | ignore whitespace

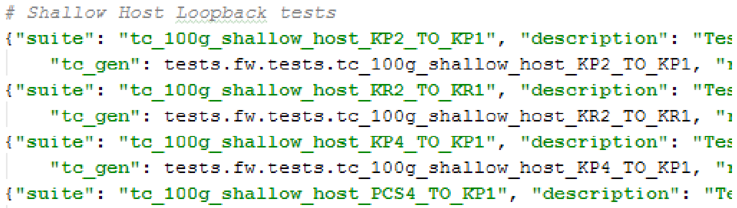
PORS-90 FW: Get shallow host loopback working!
lab@ott-lab-2000
2018-04-04 bfb509d997cc4f82629ee680f2a3c891c367bfd5

```
PORS-90 FW: Get shallow host loopback working!
```

486 ▪▪▪▪▪ **changed files**

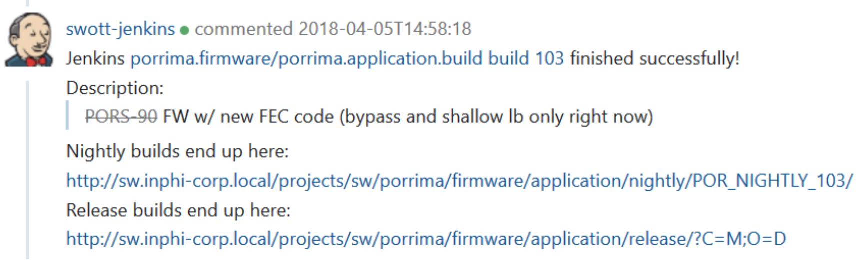
- firmware/application/inc/por_fec.h
- firmware/application/src/por_fec.c
- firmware/application/src/por_hrx_fw.c
- firmware/application/src/por_rules.c
- lab/users/dlinnington/shlb.c

#3: Write SW regression tests



```
# Shallow Host Loopback tests
{"suite": "tc_100g_shallow_host_KP2_TO_KP1", "description": "Tes
    "tc_gen": tests.fw.tests.tc_100g_shallow_host_KP2_TO_KP1, "r
{"suite": "tc_100g_shallow_host_KR2_TO_KR1", "description": "Tes
    "tc_gen": tests.fw.tests.tc_100g_shallow_host_KR2_TO_KR1, "r
{"suite": "tc_100g_shallow_host_KP4_TO_KP1", "description": "Tes
    "tc_gen": tests.fw.tests.tc_100g_shallow_host_KP4_TO_KP1, "r
{"suite": "tc_100g_shallow_host_PCS4_TO_KP1", "description": "Te
```

#4: Build new RC, track in Youtrack issue



swott-jenkins ● commented 2018-04-05T14:58:18

Jenkins porrima.firmware/porrima.application.build build 103 finished successfully!

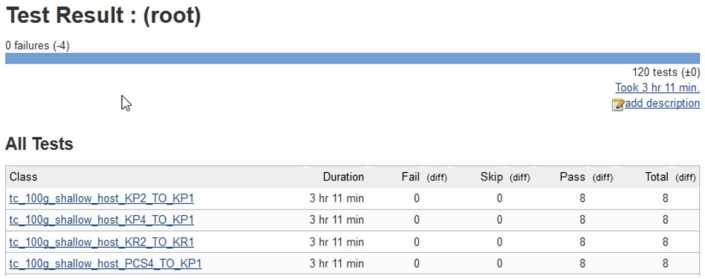
Description:

> ~~PORS-90~~ FW w/ new FEC code (bypass and shallow lb only right now)

Nightly builds end up here:
http://sw.inphi-corp.local/projects/sw/porrima/firmware/application/nightly/POR_NIGHTLY_103/

Release builds end up here:
http://sw.inphi-corp.local/projects/sw/porrima/firmware/application/release/?C=M;O=D

#5: Run tests in Jenkins against RC



Test Result : (root)

0 failures (-4)

120 tests (±0)
Took 3 hr 11 min.
add description

All Tests

Class	Duration	Fail (diff)	Skip (diff)	Pass (diff)	Total (diff)
tc_100g_shallow_host_KP2_TO_KP1	3 hr 11 min	0	0	8	8
tc_100g_shallow_host_KP4_TO_KP1	3 hr 11 min	0	0	8	8
tc_100g_shallow_host_KR2_TO_KR1	3 hr 11 min	0	0	8	8
tc_100g_shallow_host_PCS4_TO_KP1	3 hr 11 min	0	0	8	8

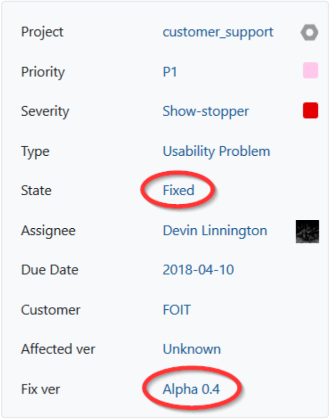
#6: Iterate through release flow, document release

1.3.21. Version 0.4.124 2018-04-18

- FEC FW is in, only supports bypass and shallow loopbacks for now

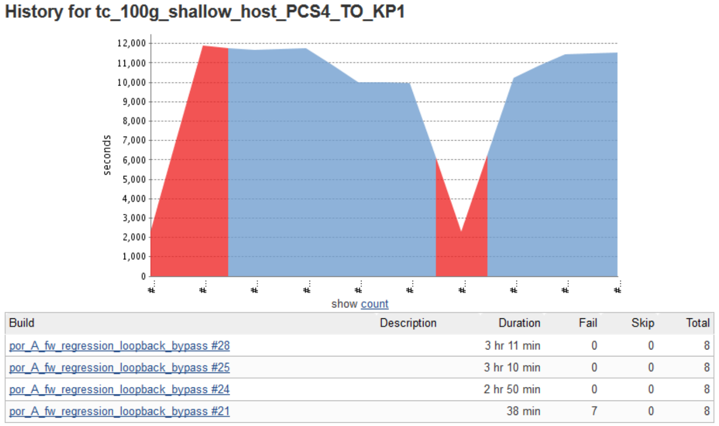
Customer Issue/Change Management

#7: Mark Youtrack issue as Fixed



Project	customer_support
Priority	P1
Severity	Show-stopper
Type	Usability Problem
State	Fixed
Assignee	Devin Linnington
Due Date	2018-04-10
Customer	FOIT
Affected ver	Unknown
Fix ver	Alpha 0.4

#8: Regress each new RC against tests

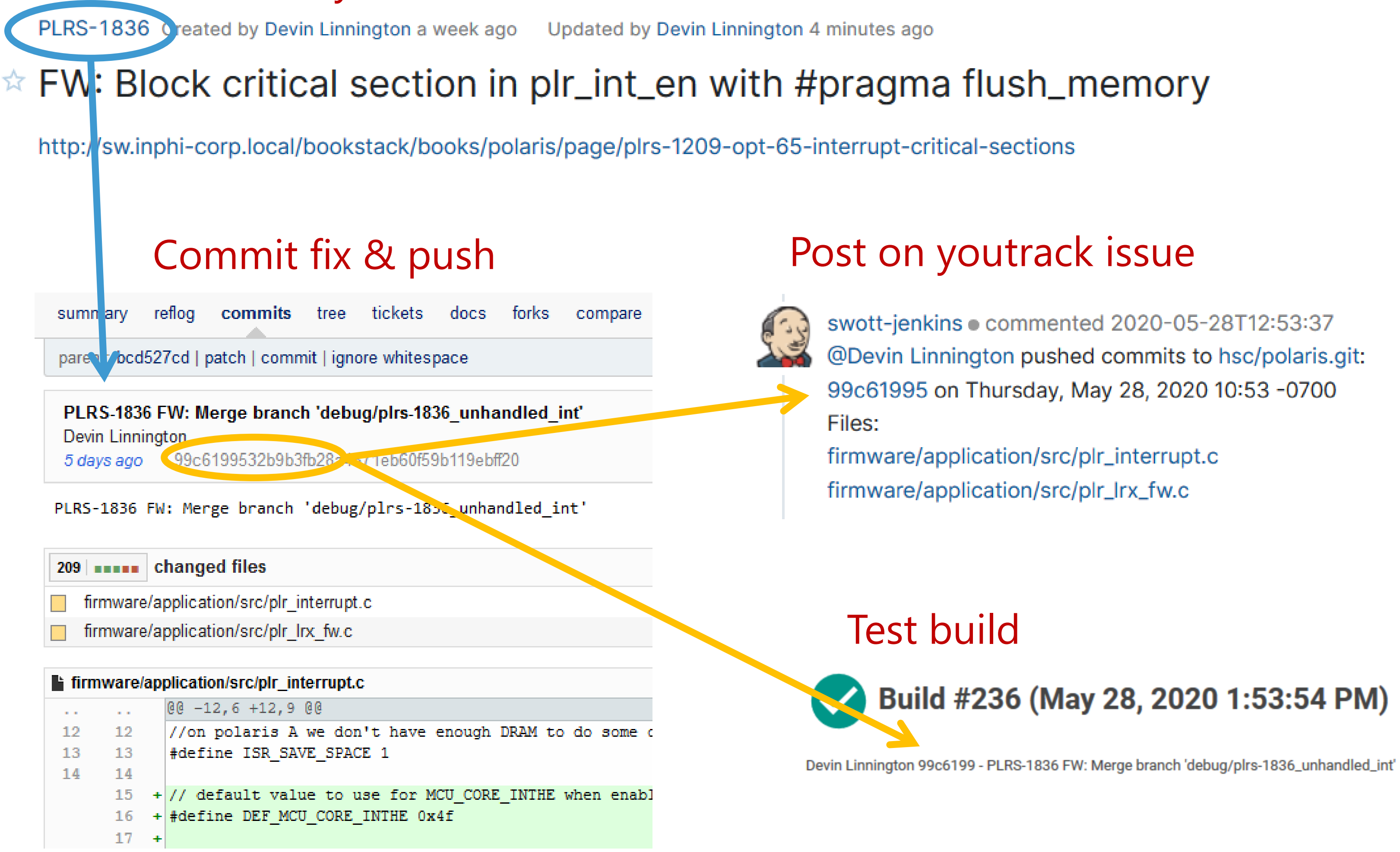


History for tc_100g_shallow_host_PCS4_TO_KP1

Build	Description	Duration	Fail	Skip	Total
por_A_fw_regression_loopback_bypass #28		3 hr 11 min	0	0	8
por_A_fw_regression_loopback_bypass #25		3 hr 10 min	0	0	8
por_A_fw_regression_loopback_bypass #24		2 hr 50 min	0	0	8
por_A_fw_regression_loopback_bypass #21		38 min	7	0	8

- Streamline development
  - Youtrack issues are reviewed prior to development; customer issues always take priority!
  - Every issue has an owner; no issues are ignored/forgotten
- Increase accountability
  - All code changes, builds, and tests are automatically reported in the Youtrack issue
  - Issue provides all the background reasoning for a change
- Increase quality
  - Regression testing catches majority of problems prior to release

Jenkins/git/Youtrack Commit Integration



Make a new youtrack issue

PLRS-1836 Created by Devin Linnington a week ago   Updated by Devin Linnington 4 minutes ago

☆ FW: Block critical section in plr_int_en with #pragma flush_memory

http://sw.inphi-corp.local/bookstack/books/polaris/page/plrs-1209-opt-65-interrupt-critical-sections

Commit fix & push

summary reflog commits tree tickets docs forks compare

parent bcd527cd | patch | commit | ignore whitespace

PLRS-1836 FW: Merge branch 'debug/plrs-1836_unhandled_int'

Devin Linnington

5 days ago 99c6199532b9b3fb28a[illegible]1eb60f59b119ebff20

```
PLRS-1836 FW: Merge branch 'debug/plrs-1836_unhandled_int'
```

209 ■■■■■ **changed files**

- firmware/application/src/plr_interrupt.c
- firmware/application/src/plr_lrx_fw.c

firmware/application/src/plr_interrupt.c

```
@@ -12,6 +12,9 @@
//on polaris A we don't have enough DRAM to do some c
#define ISR_SAVE_SPACE 1

+// default value to use for MCU_CORE_INTHE when enabl
+#define DEF_MCU_CORE_INTHE 0x4f
+
```

Post on youtrack issue

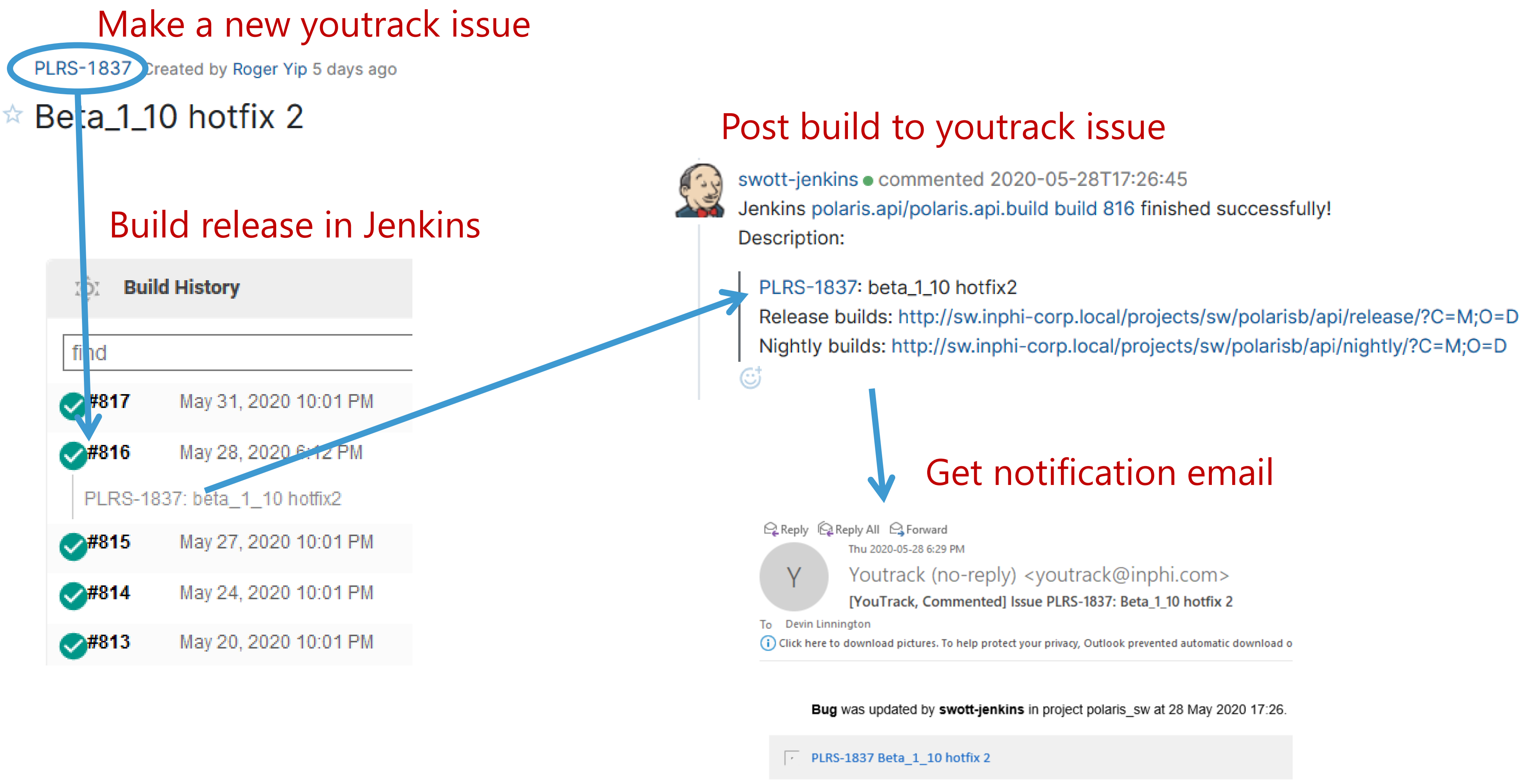
swott-jenkins ● commented 2020-05-28T12:53:37
@Devin Linnington pushed commits to hsc/polaris.git:
99c61995 on Thursday, May 28, 2020 10:53 -0700
Files:
firmware/application/src/plr_interrupt.c
firmware/application/src/plr_lrx_fw.c

Test build

Build #236 (May 28, 2020 1:53:54 PM)

Devin Linnington 99c6199 - PLRS-1836 FW: Merge branch 'debug/plrs-1836_unhandled_int'

Jenkins/Youtrack Build Integration



Make a new youtrack issue

PLRS-1837 Created by Roger Yip 5 days ago

☆ Beta_1_10 hotfix 2

Build release in Jenkins

Build History

find

✓ #817	May 31, 2020 10:01 PM
✓ #816	May 28, 2020 6:12 PM
PLRS-1837: beta_1_10 hotfix2	
✓ #815	May 27, 2020 10:01 PM
✓ #814	May 24, 2020 10:01 PM
✓ #813	May 20, 2020 10:01 PM

Post build to youtrack issue

swott-jenkins ● commented 2020-05-28T17:26:45

Jenkins polaris.api/polaris.api.build build 816 finished successfully!

Description:

> PLRS-1837: beta_1_10 hotfix2
> Release builds: http://sw.inphi-corp.local/projects/sw/polarisb/api/release/?C=M;O=D
> Nightly builds: http://sw.inphi-corp.local/projects/sw/polarisb/api/nightly/?C=M;O=D

Get notification email

Thu 2020-05-28 6:29 PM

Y Youtrack (no-reply) <youtrack@inphi.com>

[YouTrack, Commented] Issue PLRS-1837: Beta_1_10 hotfix 2

To Devin Linnington

ⓘ Click here to download pictures. To help protect your privacy, Outlook prevented automatic download o

Bug was updated by **swott-jenkins** in project polaris_sw at 28 May 2020 17:26.

PLRS-1837 Beta_1_10 hotfix 2

Questions?

Inphi®

Think fast.