

PRODUCT BRIEF

Pinetree

Integrated Algorithm Development Environment



Software for the Product Engineer

Pinetree is an ultra-capable development environment that allows you to easily and seamlessly develop and verify all your high-speed digital and mixed-signal algorithms. Designed for users with widely varying backgrounds and expertise, it offers an extremely intuitive interface simultaneously with infinitely extensible capability. From DC to GHz, from time-domain to frequency-domain, and from digital storage to video broadcast, Pinetree is the ideal development environment for the most specialized product engineering teams.

KEY FEATURES:

- **Rapid component class creation:** click and drag Python classes without any programming
- Quick result access: view results without leaving the Pinetree Procedure Editor pane
- Advanced auto-completion editor: discover all available methods without reading documentation

KEY BENEFITS:

- No software engineering expertise: protocol-aware functions with specifications-based syntax
- Advanced automation: leverage the full power of Python and the Introspect Technology hardware
- Interoperability: plug-in with multiple software languages and operating systems platforms

Pinetree Graphical User Interface

	> DEVICES	Components	csilmagePattern1		<	Procedure	0
	> GENERAL	Collection in the second secon	timeUnits	nanosecond	\sim	<pre>1 mipiGenerator1.setup()</pre>	
	 MIPI MipiClockConfig 		imageFiles	[IntrospectLogo.png]	$\overline{\mathbf{v}}$		
	CommandPattern		imageFormat	CSI_RGB888	$\overline{\mathbf{v}}$		
	CommandPattern		gaussianBlurRadius	0			
		+†+ params1	lineTimeMode	lineTimeTotal	$\overline{\mathbf{v}}$		
	ScilmageInterleaverPat		horizLineTime	30000.0			
	SilmagePattern		frameBlankingMode	frameRate	$\overline{\mathbf{v}}$		
	CustomPattern		frameRate	30.0			
	MipiGenerator		lineNumbering	disabled	\sim		
	JitterInjection		wantFrameNumbering	False	$\overline{\mathbf{v}}$		
	+++ CphyParams		enableCsiEpd	False	\checkmark		
	PatternSequence		timeUnits %	(1.0		
	DataCapturePlayer		Select the time units to use for this component. Pixel time represents the transmission period of a single pixel.				
	> UTILITY						
		\sim log					
Å							
7							
-							

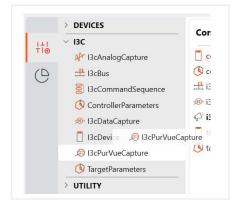


PRODUCT BRIEF

Pinetree

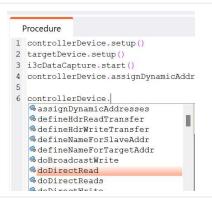
Workflow Features

COMPONENT CREATION



Click and drag components from the Component Creator tray to instantiate them in your Pinetree Procedure

INTELLIGENT EDITING



Discover component methods as you type. Simply enter the component name and instantly receive a list of available methods

RESULT QUICK ACCESS

	> Run_2020-01-29_0733	
⊥ ⊺ ⊕	> Run_2020-01-29_0738	Com
TI⊕	> Run_2020-01-29_0744	🗠 ale
()	> Run_2020-01-29_0747	0100 ale
~	> Run_2020-01-29_0759	ER CA
	~ Run_2020-01-29_0803	BER CS
	✤ DCALoopbackAnalogCapt	A DC
	DCALoopbackBertMeasur	BER DC
	DCALoopbackBertScan	⊡ DC
	XX DCALoopbackEyeScan	XX DC
	txComplianceTestExample	DC
		E DC

Click on the Result Quick Access tray to quickly view the most recent results of executing your Pinetree Procedure

Debugging and Traceability Features

SHOW-AS-TEST

	> Run_2020-01-29_0733	Com
⊥ ⊺ ⊕	> Run_2020-01-29_0738	com
	> Run_2020-01-29_0744	💌 ale
()	> Run_2020-01-29_0747	oin ale
~	> Run_2020-01-29_0759	BER CA
	~ Run_2020-01-29_0803	SER CST
	DCALoopbackAnalogC. Sho	w Folder
	III DCALoopbackBertMea: Sho	ow As Test
	DCALoopbackBertScan No	tes
	XX DCALoopbackEyeScan Ref	resh Result
	🖹 txComplianceTestExam Dol	oto Rosult

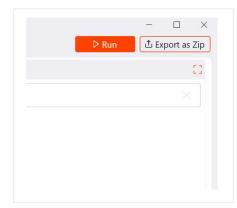
Restore any instrument and deviceunder-test state even if you have not saved your most recent Pinetree Procedure

INTELLIGENT LOGGING

\sim log			
тоддтид	LESL	uala	101
logging	test	data	182
logging	test	data	183
logging	test	data	184
logging	test	data	185
logging	test	data	186
logging	test	data	187
logging	test	data	188
logging	test	data	189
logging	test	data	190
logging	test	data	191
logging	test	data	192
logging	test	data	193
logging	test	data	194
1	* *	-1-+-	105

Activity logging is stored on every execution of a Pinetree Procedure. What's more, you can customize your own logging features

EASY SHARING



With the new Export as Zip button, you can share your entire Pinetree project without needing to locate files in your operating system



PRODUCT BRIEF

Pinetree

Pinetree Procedure Authoring Features

GRAPHICAL ATTRIBUTE EDITOR

	Components 🔅	txDQDQSJitterTestB	ertScan	
	🐣 aiertunanneillist	timeUnits	picosecond	\sim
5	AlertDebugDigitalCapture CAfrainingBertMeasure CAfrainingBertMeasure DCALoopbackAnalogCa DCALoopbackBertScan ZOCALoopbackEyeScan DCALoopbackEyeScan DCALoopbackTest DCATM DDS_Test_Suite DPS_Test_Suite	rxChannelList	rxDQDQSChannelList	\sim
		bertDurationInBits	1e+06	
		startPhase	-120.0	
		endPhase	120.0	
		phaseOffsets	0	
		onlyDoSetupOnce	True	\sim
		patternSync	PATSYNC_strobeSync	\sim
Ĩ		measurementMode	allTransitions	\sim
d		wantAnalysis	True	\sim
Ē		saveResults	True	\sim
	ddrRcdCommandPattern	wantResultImages	False	~

For users who do not want to programmatically adjust component attributes, Pinetree enables graphical editing of components and their properties

PROCEDURE EDITOR

Procedure	generateData	createFinalReport	simulateData
8 print("A	ssigning dynamic	addresses")	
9 daaTable	= masterDevice.	assignDynamicAddre:	sses()
LO			
ll # Print	DAA Table		
12 pif (daaT	able == None):		
13 slvA	ddresses = []		
4 pelse:			
15 slvA	ddresses = daaTa	ble.keys()	
l6 print("M	aster DAA Table	(%d entries)" % ler	n(slvAddresses))
17pfor slvA	ddr <mark>in slvA</mark> ddres	ises:	
18 prin	t("slvAddr=%d, %	s" 🖇 (slvAddr,daaTa	able[slvAddr]))
l9 print("	")		
20			
21 # Readin	g Slave Register	s	
22pfor slvA	ddr in slvAddres	ses:	
23 # Di	rect Read 'GETBO	CR'	
	- / "Deading Due C	househouisties Des-	takan Kan alwadanda

The Pinetree Procedure Editor is a full-featured Python interpreter, featuring color coding, automatic indentation and much more

Eile Edit IESP Tools Besults Help L Export as Zip Procedure sideBandBusReadDword sideBandBusWriteByte 141 T 100 channelNum, pageNum, regNum, wrData C # Performs a single byte(8 bits) write # to the RCD register specified by the argu # All arguments should be integers. Pinetree (v 23.3.b5) - DDR5 FullDIMM testSuite (SVSC 16C12G DDR Eile Edit IESP Tools Results Help 🖞 Export as Zip Procedure sideBandBusRe 0 <= regNum <= 0x7F, "RW number in 0 <= pageNum <= 0x04, "PageNum in Americinanneiust Senter Chiraining 141 T 100 channelNum, pageNum, regNum, wrData ssp = getIespInstance() esp.setI2cRegAddrMode(1) # Single byte addressing e CATrainingBertMeasure... rms a single byte(8 bits) write > RCD register specified by the argum rguments should be integers. CSTrainingBertMeasure. 12cDeviceAddress = 0x5F # I2C adress for HID=111 (10) iesp.setI2cDeviceAddress(i2cDeviceAddress) DCALoopbackAnalogCa. DCALoopbackBertMeas. 52 DCALoopbackBertScar III DCALoopbackEveScan Suite (SV5C 16C12G DDR $\square \rightarrow$ i2cPacket DCALoopbackTest Edit JESP Jools Results Help ▷ Run ① Export as Zip DCATM E DCSTM txDODOSJ DDR5 Test Suite 0x5F # I uddress (12) E ddrBodCom rxChannelLis rxDQDQS defaultSync noSync G us a single byte(8 bits) write RCD register specified by the startPhase CSTrainingBertMe -120.0 endPhase phaseOffsets onlyDoSetupOn PCALoopbackAnalogCa 20.0 ~ LOG 21 i2cPacket = [byted Serial # Pw revision: Personality DCALoopbac True globalClockConfig # SideBand bus of iesp.writeI2cRed XX DCALoopbackEyeScan patternSync PATSYNC str IpbkDataCh PMICEnable (Ch.A) 自 powerOr 自 QCSTM G DOR5 Test Suit LOG On DOLoophackDigitalCan timeUnits % 20 21 i2cPacket = [byte gNum, wrData] 22 print(i2cPacket) PMICEnabl Ð ~ LOG SV5C_16C12G_DDR No

With focus modes, you can optimize your own work style. You can maximize the Pinetree Procedure Editor and minimize everything else (top left). Or, you can focus on only the editor and instantiated components (middle). Or, you can expand the Attributes Editor to manually manipulate the component properties (bottom right)

Focus Modes