## \* CCS3.3 FFT 알고리즘 Project

- 1. TI제공 예제파일을 설치합니다.
- . SPRC624.zip 예제프로그램
- 2. 본 예제 에서는 당사에서 제공한 아래 라이브러리를 사용한다. 기존 FFT라이브러리에 함수 추가 후 FLASH 메모리에서 실행 하도록 하였습니다. C:₩tidcs₩c28₩C28x\_FPU\_Lib



3. Setup CCStudio v3.3을 실행 합니다.



4. 프로그램 및 디버거에 사용할 장치를 선택 한후 Add버튼을 클릭 합니다.

🌍 Code Composer Studio	Setup				
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>H</u> elp					
System Configuration	Augilable Festery Re	rda	Family	Platform	Endianness
	Available 1 actory bo		С28хх 💌	xds510usb 💌	All 🗾
🚇 My System	F2801 XDS510USE	Emulator	C28xx	xds510usb,	*
	F2806 XDS510USB	Emulator	C28xx	xds510usb	*
	F2808 XDS510USB	Emulator	C28xx	xds510usb	*
	F2810 XDS510USB	Emulator	C28xx	xds510usb	*
	F2811 XDS510U9B	Emulator	C28xx	xds510usb,	*
	F2812 XDS51005B	Emulator	C28xx	xds51Uusb	*
	F28332 XDS510DSE	Emulator	C28XX	XdS5IUUSD	*
		Emulator	C20XX	xassiouso,	*
		Emulator	CZOXX	Xassiousp,	*
	🔤 Factory Boards	🔤 Custom Boards 🗌	Create Board	J	
Save & Quit Remove	< Add	<< Add Multiple			
Drag a device driver to the left to	, add a board to the syst	tem,			

🐉 Code Composer Studio	Setup	
<u>File E</u> dit <u>V</u> iew <u>H</u> elp		
System Configuration	Available Pro	Driver Location
I My System I F28335 XDS510USB Emu I Cpu_0	★ TMS320F2400 ★ TMS320F2800 ★ BYPASS	C:₩CCStudio_v3,3₩drivers₩s, C:₩CCStudio_v3,3₩drivers₩s,
<	📷 Factory Boa	rds 📴 Custom Boards 🍖 Cr 📢
Save & Quit Remove	<< Add	<< Add Multiple
Select the system node to add a	new board to the s	system configuration,

5. My System에서 F283335 XDS510USB Emu를 선택후 Start Code Composer Studio를 실행 합니다.

🐌 Code Composer Studio	Setup		
<u>File E</u> dit <u>V</u> iew <u>H</u> elp		_	
<u>R</u> emove All Reventes Court Costiennetics	Ctrl+N	roc	Driver Location
Nevert to Saved Configuration Save	Ctrl+O Ctrl+S	0F2800	C:₩CCStudio_v3,3₩drivers₩s
<u>I</u> mport, <u>E</u> xport,			
Start Code Composer Studio			
E <u>x</u> it	Alt+F4		
Save & Quit Remove	Fact	ory Boar	rds 🚾 Custom Boards 🌨 Cr. া

## \* CCS3.3 DSP FFT Project 시작

1. Setup CCStudio v3.3 이나 CCSStudio3.3을 실행 합니다.



2. 아래와 같이 Project를 오픈 합니다.(Project->Open)

휋 /F28335 XDS510	OUSB Emulator/cpu_0 - TMS320C28xx	- Code Composer Studio - Not Connected
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project Debug GEL Option Profil	e <u>T</u> ools D <u>S</u> P/BIOS <u>W</u> indow <u>H</u> elp
12 🚅 🖬 🕹	<u>N</u> ew	🖵 🖌 🖓 🐐 🦬 😘 🖨 😽
	<u>O</u> pen	
	Us <u>e</u> External Makefile	🖽 🏝   🕛 🕸   🗣 🔑
	Export to Makefile	
凝의 ( 💭 📋 🗍	Add Files to Broject	

Project Open	28	
찾는 위치([):	🔁 fft_test 💽 🖛 🖽 -	
test, CS_		티 ASH에서 식행되는
		프로젝트
파일 이름( <u>N</u> ):	test 열기( <u>0</u> )◀	버튼 클릭
파일 형식( <u>T</u> ):	Project Files (*,pjt) 💌 취소	
	도움말( <u>H</u> )	

	Projects Source 파일 구성			
	Files			
	🖻 🧰 GEL files			
	🖻 🔄 Projects			
	🖻 🚔 test pjt (Debug)			
	Dependent Proj	ects		
	Documents	<i>a</i>		
		tig		
	Generated Files	3		
	⊡ Include			
	—————————————————————————————————————		인터럽트	및 CPU관련 초기화
		VUULUAI, ASIII °odoStortPropoblioc		
		JoueStartDrahtth, as StobolVoriobleDofe	 ~	
	I DSF2000X_0 I DSF2000X_0	uobarvanabiebeis. .ci c	6	
	→ DSF2003×15	ici, c IeNolau aem		
	tost c	isbelay,ashi	메이고르	고래
		ters nonBIOS cmd	에한 프로	
		nund cmd 🗕 🚽		
4. F	Projects Option 구성(Project->Build	d Option)	FLASH-	RON LINKER COMMAND FILE
	Build Options for test.pjt (Deb	oug)		<u>? ×</u>
	General Compiler Linker DspB	iosBuilder   Link Order		
	_g -pdsw225 -fr"\$(Proj_dir)₩Debu	g" -i"\	mmon₩includ	e" 📉
	-i"₩₩DSP2833x_headers₩inclu  d"_DEBUG" -d"LARGE_MODEL" -n	ude" -i",,₩,,₩C28x_FPU_ nl -mt -v28float_suppo	Lib₩include" · rt=fpu32	
	Category: Preprocessor	2-4- / N.J		
	Category: Preprocessor Basic Include Search F Advanced Dre Define Surek	Path (-i):[ude),,₩,,₩C28;	<_FPU_Lib₩in	clude 🔶 FPU Library Include 추기
	Category: Preprocessor Basic Advanced Feedback Files	Path (-i): ude),,₩,,₩C28; bol (-d): LDEBUG;LARGE	<_FPU_Lib₩in( _MODEL	clude 🔶 FPU Library Include 추기
	Category: Preprocessor Basic Advanced Feedback Files Assembly Parser Pre-Define Symbo	Path (-i): ude),,₩,,₩C28; bol (-d): _DEBUG;LARGE I (-u):	<_FPU_Lib₩in¢ _MODEL	clude 수 FPU Library Include 추기
	Category: Preprocessor Basic Advanced Feedback Files Assembly Parser Preprocessor Undefine Symbo Preprocessing: Continue with	Path (-i): ude;,,₩,,₩C28; bol (-d): LDEBUG;LARGE I (-u): None n Compilation (-ppa)	<_FPU_Lib₩ind _MODEL _	clude ← FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Undefine Symbo Preprocessing: Continue with	Path (-i): ude;.,₩,,₩C28; bol (-d):  _DEBUG;LARGE I (-u):  None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	clude 수 FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics Preprocessor Undefine Symbo Preprocessing: Continue with	Path (-i): ude;,,₩,,₩C28; bol (-d):  _DEBUG;LARGE I (-u):  None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	Clude + FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics Preprocessor Diagnostics	Path (-i): ude;.,₩,,₩C28; bol (-d):  _DEBUG;LARGE I (-u):  None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	Clude ← FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Diagnostics Preprocessor Undefine Symbo Preprocessing: Continue with	Path (-i): ude),,₩,,₩C28; bol (-d):  _DEBUG;LARGE I (-u):  None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	Clude ← FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Diagnostics Preprocessor Undefine Symbo Preprocessing: Continue with	Path (-i): ude),,₩,,₩C28; bol (-d):  _DEBUG;LARGE I (-u):  None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL	clude + FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics Preprocessing: Continue with	Path (-i): ude;,,₩,,₩C28; bol (-d):  _DEBUG;LARGE I (-u):  None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	clude + FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics Preprocessor Continue with	Path (-i): ude;,,₩,,₩C28; bol (-d):DEBUG;LARGE I (-u): None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	Clude + FPU Library Include 추기
	Category: Basic Advanced Feedback Files Assembly Parser Preprocessor Diagnostics Preprocessor Continue with	Path (-i): ude;,,₩,,₩C28; bol (-d):DEBUG;LARGE I (-u): None n Compilation (-ppa)	<_FPU_Lib₩in( _MODEL _	Clude + FPU Library Include 추기

Build Option	s for test.pjt (Debug)	? 🔀
General Con -c -ecode_s -w -x -i"₩ I"rts2800_fpu	npiler Linker DspBiosBuilder Link Order tart -m", WDebugWtest, map" -o", WDebugWtest, out" ., WDSP2833x_commonWlib" -i",, W,, WC28x_FPU_Lib 32, lib" -l"1Qmath, lib" -l"C28x_FPU_Lib_Beta1, lib"	-stack0x380 ▲ ₩lib¨ -
Category: Basic Libraries Advanced	Libraries ✓ Exhaustively Read Libraries (-x) Search Path (-i): [3833x_common₩lib),,₩,,₩C28x Incl, Libraries (-I):]_fpu32,lib;IQmath,lib;C28x_FPU	A_FPU_LibWlib U_Lib_Beta1,lib ア FPU Library 추가
	확인 취소	도움말

## - CCS3.3 DSP Program 실행

1. JTAG 및 에뮬레이터를 연결 합니다.

😻 /F28335 XDS510USB	Emulator/cpu_0 -	TMS320C28xx - Co	de C
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>P</u> roject	Debug <u>G</u> EL Option	P <u>r</u> ofile <u>T</u> ools D <u>S</u> P/I	BIOS
웥 🚅 🔚 👗 🖻 💼	<u>B</u> reakpoints		12
test, pjt 🗾 💽	Assembly/Source S Step Into	tepping F11	2
日本	Step <u>O</u> ver Step O <u>u</u> t	F10 Shift+F11	
(*)	<u>Bun</u> <u>H</u> alt <u>A</u> nimate Run <u>F</u> ree <u>L</u> ow Power Run Run to Cursor Ret PC to Cursor R <u>e</u> start <u>G</u> o Main Multiple Operation,	F5 Shift+F5 Alt+F5 Ctrl+F5 Ctrl+Alt+F5 Ctrl+F10 Ctrl+Shift+F10 Ctrl+Shift+F5 Ctrl+M	
<ul> <li>※</li> <li>※</li> </ul>	Adva <u>n</u> ced Resets Reset CPU Reset Emulator Halt on Reset	► Ctrl+R Ctrl+Shift+R	
□ □	<u>Connect</u> Restore <u>D</u> ebug State	Alt+C ┥	

2. 내부램 으로 프로그램을 실행할 경우 아래와 같이 설정 합니다.(Option->Customize)

Customize 🛛 🛛 🗙	
Debug Properties   Directories   Color   Keyboard   Program/Project/CIO   Control Window Dist	
Program Load         ✓ Perform verification during Program Load         ✓ Load Program After Build         ✓ Do Not Set CIO Breakpoint At Load         ✓ Do Not Set End Of Program Breakpoint At Load         ✓ Disable All Breakpoints When Loading New Programs	_ 체크
Project         Image: Open Dependent Projects When Loading Projects         Image: Do Not Scan Dependencies When Loading Projects         Image: Auto-save Projects Before Build         Image: Open Project Window On Startup	
CIO Maximum # of lines: 256 (may require restart)	
<u>확인</u> 취소 적용( <u>A</u> ) 도움말	

## 3. 컴파일 하기(Project->Rebuild All)

휋 /F28335 XDS51	LOUSB Emul	ator/cpu_0	- TMS32	)C28xx -	Code C
<u>F</u> ile <u>E</u> dit <u>V</u> iew	Project D	ebug <u>G</u> EL	<u>O</u> ption	P <u>r</u> ofile	Tools
<sup>™</sup>	<u>N</u> ew <u>O</u> pen Us <u>e</u> Exte	rnal Makefi	le		i ##
응산         월           (%)         (%)           (%)	E <u>x</u> port to <u>A</u> dd File <u>S</u> ave <u>C</u> lose	o Makefile s to Project			
r∰ L <mark>@</mark> f.	So <u>u</u> rce (	Control		I	
??)         □	Compile <u>B</u> uild	<u>F</u> ile	i I	Ctrl + F 7 F <b>7</b>	
	<u>R</u> ebuild	All			



에러 확인

4. FLASH에 프로그램 하기

۰ 😻	/F2833	5 XDS	51 OUSE	Emula	tor/c	ou _0 -	TMS32	JC28xx	- Code C	ompose	r Studio	- Not
<u> </u>	<u>E</u> dit	<u>V</u> iew	<u>P</u> roject	<u>D</u> ebug	<u>G</u> EL	<u>O</u> ption	Profile	<u>T</u> ools	D <u>S</u> P/BIOS	<u>W</u> indow	<u>H</u> elp	
- <b>*</b>	🖻 F	ЦX	n Ri	$ \alpha   \alpha $				F28x	x On-Chip F	lash Progr	ammer	<b>N</b> ?
							-	Data	Converter S	upport		· ·

\* 아래 CLOCK 설정 메뉴를 사용자에 맞게 설정 합니다.

<b>Clock Configuration</b>		×
OSCCLK	30	OK
DIVSEL:	/2 💌	Cancel
PLLCR Value:	10 -	
SYSCLKOUT	150,0000	

\* API Interface 파일을 등록 합니다.

Flash Programmer Settings			
Select DSP Device to Program	Options ↓ Load Symbols ↓ Display Tooltips ↓ Display Diagnostics ↓ Save Clock Settings	OK Cancel Help	
Select version of Flash API Interface			
		Browse	

열기				? 🗙
찾는 위치([):	28335	•	+ 🗈 💣 📰+	
FlashAPIInterfac	ce28335V1_01, out			
파일 이름( <u>N</u> ):	FlashAPIInterface28335V2_10,out		<b>_</b>	1(0)
 파일 형식( <u>T</u> ):	COFF Files (*,out)		▼ <sup>*</sup>	1소

Flash Programmer Settings							
Select DSP Device to Program	Options Load Symbols Display Tooltips Display Diagnostics Save Clock Settings	OK Cancel Help					
Select version of Flash API Interface							
	#Algorithms#20000#FlashAr	Interfacezo:					

On-Chip Flash Programmer Clock Configuration OSCCLK 30 DIVSEL: /2 ▼ PLLCR Value: 10 ▼ SYSCLKOUT 150,0000 Code Security Password Key 7 FFFF Key 6 FFFF Key 5 FFFF	-Erase Sector Selection ✓ Sector A: (338000-33FFFF) ✓ Sector F: (310000-317FFF) ✓ Sector B: (330000-337FFF) ✓ Sector G: (308000-30FFFF) ✓ Sector C: (328000-32FFFF) ✓ Sector H: (300000-307FFF) ✓ Sector D: (320000-327FFF) ✓ Sector I: (3DA000-3DBFFF) ✓ Sector E: (318000-31FFFF) ✓ Sector J: (3DB000-3DBFFF) ✓ Sector E: (318000-31FFFF) ✓ Sector J: (3DB000-3D9FFF) ✓ Depretion Please specify the COFF file to D:\#lang\cupulaza\dsp28335\project\dtft_test\Debug Browse, ✓ Erase, Program, Verify O Depletion Recovery		Browe 중 파일을 선
Key 4       FFFF         Key 3       FFFF         Key 2       FFFF         Key 1       FFFF         Key 0       FFFF         Unlock       Lock         Program Password       Flack Braggemeer Settinge	C Erase Only C Frequency Test Program, Verify Register: GPAMux Program Only Pin: GPIO0 (A) Verify Only Flash Random Wait State: 15 Flash Page Wait State: 15 OTP Wait State: 31 C Load RAM Only Flash+OTP: Execute Operation Help		이peratior 합니다. * TI 실행 현재 작 ₩deb
		-	

- Browe.. 창에서 파일을 선택후 Excute Operation탭을 실행 합니다.
- \* TI 실행 파일은 \*.OUT로 현재 작업 디렉토리 ..∀debug₩ 에 있습니다.



Clock Configuration       Frase Sector Selection         OSCCLK       30         DIVSEL:       /2         Y       Sector A: (338000-33FFFF)         VSEL:       /2         PLLCR Value:       10         SYSCLKOUT       150,0000         Code Security Password       ✓ Sector B: (310000-32FFF)         Key 7       FFFF         Key 5       FFFF         Key 4       FFFF         Key 3       FFFF         Key 4       FFFF         Key 1       FFFF         Key 2       FFFF         Key 1       FFFF         Key 1       FFFF         Key 0       FFFF         Vority Only       Piase Social State:         Y       Program Password         Vority Only       Flash Random Wait State:         Y       Flash Page Wait State:         Y       Flash Page Wait State:         Y       Flash+OTP:
Flash Programmer Settings



5. 프로그램을 로딩후 Debug 탭에서 Go Main 기능을 실행 합니다.

6. TIME 파형 생성 부분에 Break(F9 KEY)를 설정후 RUN(F5 키)시켜 파형을 확인 합니다.

// FFT 관련 float32 InBuffer[FFT_SIZE]; // TIME 과형 입력 버피 float32 OutBuffer[FFT_HALF_SIZE]; // FFT 결과 버피 float32 TwiddleBuffer[FFT_SIZE]; // FFT 한수 참고용 테이블 float32 MagBuffer[FFT_HALF_SIZE]; // FFT 친구 테이블 연산하여 저장 하는 버피 float32 Fft_ResultBuffer[FFT_HALF_SIZE]; // FFT 최종 연산 결과 저장 RFFT_F32_STRUCT fft; // FFT 사용 구조체
//
<pre>void main(void) {     unsigned char ch;     unsigned int i,k;     float theta,dataf,step_f1 = 1.0 / (float)FFT_SIZE;</pre>
cpu_setup(); // H/W 초기화 및 관리
// FFT 之 7] 基         fft.InBuf = InBuffer;       // Input data buffer         fft.OutBuf = OutBuffer;       // FFT output buffer         fft.CosSinBuf = TwiddleBuffer;       // Widdle factor buffer         fft.FFTSize = FFT_SIZE;       // FFT length         fft.FFTStages = FFT_STAGES;       // FFT Stages         fft.MagBuf = MagBuffer;       // Magnitude buffer
theta = 2.0 * M_PI / (float)FFT_SIZE;
\$ RFFT_f32_sincostable(&fft); // Initialize twiddle buffer while(1){ if(com1_rx_usi != com1_rx_udi){ // 수신 김사 ch = com1_recv_data[com1_rx_udi++]; // ring 버과 데이터 로드 com1_putch(ch); // 데이터 송신
「 RFFT_f32u(&fft); // FFT 함수 RFFT_f32_mag(&fft); // FFT 결과 연산 fft.MagBuf[n] = sqrt(real^2+image^2) // RFFT_f32_mag_non_sqrt(&fft); // FFT 결과 연산 fft.MagBuf[n] = (r[n]^2+I^2) // - RFFT f32 mag_ 사용시

7. TIME 파형 확인(View->Graph->Freqency..)

🐱 Graph Property Dialog		×	
Display Type	Single Time 💼	~	
Graph Title	Graphical Display		
Start Address	fft, InBuf 🔶		확인할 버퍼명
Page	Data		
Acquisition Buffer Size	2048 🔶		표시할 네이터수
Index Increment	1		
Display Data Size	2048		
DSP Data Type	32-bit floating point 🛛 🔸		왁인알 네이터 영
Sampling Rate (Hz)	1		
Plot Data From	Left to Right	≣	
Left-shifted Data Display	Yes		
Autoscale	On		
DC Value	0		
Axes Display	On		
Time Display Unit	S		
Status Bar Display	On		
Magnitude Display Scale	Linear		
Data Plot Style	Line		
Grid Style	Zero Line	_	
Cursor Mode	Data Cursor	~	
	<u>OK</u> <u>Cancel</u> <u>H</u> elp		



- 실제값을 확인(변수위에 오른쪽 마우스 누른후 -> Add to Watch Window)

<pre>void main(void) {     unsigned char ch;     unsigned int i,k;     float theta,dataf,ste         cpu_setup();</pre>	p_f1 = 1.0 ✓ (float)Fi	- - -					
<i>FFT 초기화</i> fft.InBuf = InBuf fft.OutBuf = Out	Editor •	1	Watch Window				X
fft.CosSinBuf =	Open Document		Name	Value	Туре	Radix	
fft.FFTSize = FF	View Location of InBuffer		- Ø InBuffer	0×0000A840	float[2048]	hex	-=1
fft.FFTStages =	Add to Watch Window		♦ [0]	3,0	float32	float	
fft_MagBuf = Mag	Add to watch window			2,999934	float32	float	
l incluged mag			[2]	2,999737	float32	float	
theta = $2.0 \pm M$	Mixed Mode		§ [3]	2,999407	float32	float	
$\mathbf{for}(\mathbf{i} = \mathbf{k} = 0, \mathbf{i})$	mixed mode		[4]	2,998946	float32	float	
$d_{a+af} = \frac{1}{4}$	Cut		♦ [5]	2,998353	float32	float	
uataI = I * CO	- Conv		♦ [6]	2,997629	float32	float	
rrt.inbut[1]	Basta			2,996773	fluat32	fluat	_
				2,995785	float32	float	_
RFFI_fJ2_SIncost	Select All			2,994666	float32	float	_
while(1){	Set PC to Cureor		Q [10]	2,993415	10at32	float	_
if(coml_rx_u			Q [11]	2,992033	110at32	float	_
ch = com	<u>H</u> un To Cursor			2,39052	110at32	float	_
com1_put	Togalo Software Breakpoint			2,300070	110at52	float	-
}	Toggie Soliware Dreakpoint		♥ [14]	2,3071	110at52	floot	-
RFFT_f 32u (&f	Loggle Hardware Breakpoint	L I	♦ [15]	2,300134	floot22	floot	-
RFFT_f32 mag	Bookmarke	1		2,303137	floot32	floot	-
RFFT_f32_mag	DUOKIIIAIKS		♦ [18]	2,300303	float32	float	-
// - RFFT_f32_mag X	Insert Graph		↓ [10] ▲ [19]	2 976262	float32	float	-
// 1. mag[n] = sarti	Go to		↓ [10]	2 973703	float32	float	
// 2. FFT연산 결과 치 // step_f1 = 연산			& Watch Locals	Match 1			

8. FFT VOLT 생성 부분에 Break(F9 KEY)를 설정후 RUN(F5 키)시켜 파형을 확인 합니다.

<pre>while(1){     if(com1_rx_usi != com1_rx_udi){</pre>
//FFT 항수 RFFT_f32u(&fft); //FFT 항수 RFFT_f32_mag(&fft); //FFT 결과 연산 fft.MagBuf[n] = sqrt(real^2+image^2) // RFFT_f32_mag_hawl // - RFFT_f32_mag_hawl
// 1. mag[n] = sqrt(real^2+image^2)가 계산 되어 나오므로 // 2. FFT연산 결과 계산된 값에 FFT_SIZE만큼 값이 더해지고, 한쪽면값만 나오므로 (연산결과 / FFT_SIZE) * 2.0을 해야함, 단 DC는 // step_f1 = 연산 결과를 FFT_SIZE로 나눌때 연산 시간이 걸리므로 역수를 취해 곱합
// - 주과수별 VOLT3} // Fft_ResultBuffer[0] : DC3} // Fft_ResultBuffer[1] - [FFT_HALF_SIZE] : 각 주과수별 VOLT3} // - 주과수 구하는법
// FFC_ResultBuffer[0]: (n/D SAMPLE 주과수 / FFT_SIZE) * 간 배열 포인터 for(i = 0; i < FFT_HALF_SIZE; i++){ // FFT 결과 VOLT 계산 // RFFT_f32_mag() 사용시
if(1 == U) Fft_KesultBuffer[1] = (fft_MagBuf[1] * step_f1);// result[UC] = (mag[n] / FFT_SIZE) else Fft_ResultBuffer[i] = (fft_MagBuf[i] * step_f1) * 2.0;// result[V] = (mag[n] / FFT_SIZE) * 2.0 // RFFT_f32_mag_non_sqrt() 太}&\]
<pre>// if(i == 0) Fft_ResultBuffer[i] = (sqrt(fft_MagBuf[i]) * step_f1):// result[DC] = sqrt(mag[n] / FFT_SIZE) // else Fft_ResultBuffer[i] = (sqrt(fft_MagBuf[i]) * step_f1) * 2.0;// result[V] = sqrt(mag[n] / FFT_SIZE) * 2.0 </pre>
\$ run_cnt++; // RUN-Count } }

9. FFT VOLT 파형 확인(View->Graph->Freqency..)

🐱 Graph Property Dialog		
Display Type	Single Time 📄	
Graph Title	Graphical Display	
Start Address	Fft_ResultBuffer 🚽	확인할 버퍼명
Page	Data	
Acquisition Buffer Size	1024 🗕	표시할 네이터수(TIME/2)
Index Increment	1	
Display Data Size	1024	
DSP Data Type	32-bit floating point 🛛 🚽	확인할 네이터 형
Sampling Rate (Hz)	1	
Plot Data From	Left to Right	
Left-shifted Data Display	Yes	
Autoscale	On	
DC Value	0	
Axes Display	On	
Time Display Unit	S	
Status Bar Display	On	
Magnitude Display Scale	Linear	
Data Plot Style	Line	
Grid Style	Zero Line	
Cursor Mode	Data Cursor	
-	<u>OK</u> ancel <u>H</u> elp	

Graphi	cal Display														×
1.00	-														1
0.667-															
0.333-															- (
0															
-0.333-															
-0.667															
-1.00 <u>-</u>	64.0	128	192	256	320	384	448	512	576	640	704 76	8 832	896	960	1023
(511, 3,9	519989e-8) Fi	ft_ResultBuff	er							Ti	me	Lin	uto Scale		

- 실제값을 확인(변수위에 오른쪽 마우스 누른후 -> Add to Watch Window)

// - 주파수별 VOLT과			Watch Window				×
// Fft_ResultBuffer[0] : DCZ			Name	Value	Туре	Radix	
// Fft_ResultBuffer[1] - [FFT_HAL	F_SIZE] : 각 주파주별 VOL3	i -		0×00009040	float[1024]	hex	
- アーダナ アのご日 アーダナ アのご日 			↓ [0]	7,962808e-08	float32	float	
// Fft ResultRuffer[1] - (FFT HA)	F SIZEI : (AZD SAMPLE 74	2	II)	1,0	float32	float	
<pre>for(i = 0; i &lt; FFT HALF S</pre>	IZE; i++){ // FFT 결과	2	[2]	1,0	float32	float	
// RFFT_f32_mag() /-8/			IS]	1,0	float32	float	
if(i == 0) Fft_Result	Editor	]	I [4]	9,133088e-08	float32	float	
else Fft_ResultBuffer	Open Document		§ [5]	3,522608e-08	float32	float	
<pre>// KFFT_F32_mag_non_sqrt() XF&amp;X</pre>	View Location of Eft BesultBuffer		(6)	4,626166e-08	float32	float	
II [I == U] IIL_KeSult	Add to Watch Window		I [7]	2,470452e-08	float32	float	
l l l l l l l l l l l l l l l l l l l	Quick Watch		IS]	1,053591e-08	float32	float	
<pre>run_cnt++;</pre>			🛛 🔮 [9]	2,107226e-08	float32	float	
}	<u>M</u> ixed Mode		🛛 🔮 [10]	8,119166e-09	float32	float	
}	Cut		I [11]	2,039992e-08	float32	float	
	Copy		♦ [12]	3,58206e-09	float32	float	
	Paste		💊 [13]	1,30983e-08	float32	float	
	Select All		🛛 🔮 [14]	8,633223e-09	float32	float	
			🛛 🔮 [15]	1,73171e-08	float32	float	
	Set PC to Cursor		🛛 🔮 [16]	8,678688e-09	float32	float	
	<u>R</u> un To Cursor		IT7]	1,461737e-08	float32	float	
	Toggle Software Breakpoint		IN [18]	1,84083e-08	float32	float	
	Toggle Hardware Breakpoint		IP (19)	1,254632e-08	float32	float	
			(20)	3,208877e-09	float32	float	
	B <u>o</u> okmarks •		[21]	4,246763e-09	float32	float	
	Insert Graph			2,992204e-09	float32	float	-
	Go to			4 5033370 00	1/1 100	10 1	_
	Advanced •		🚜 Watch Locals 🔗	Watch 1			