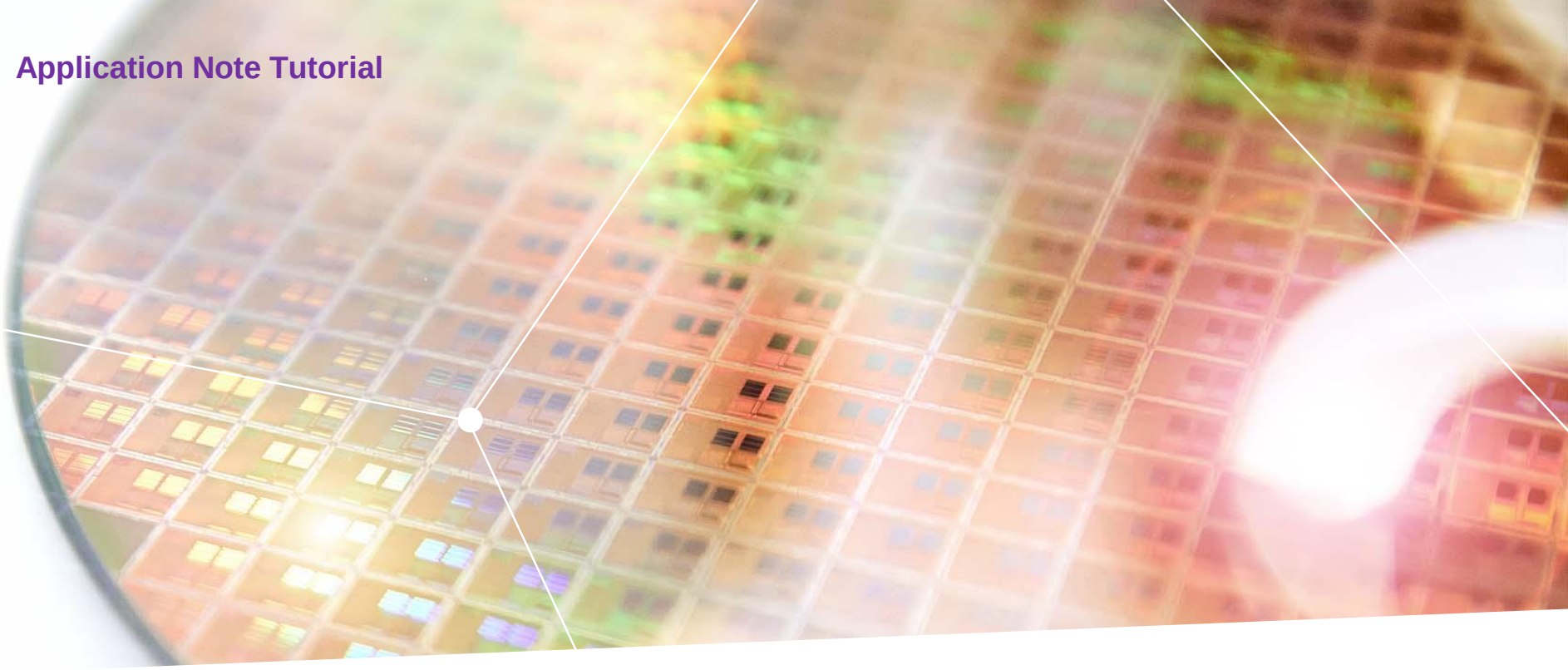
Application Note Tutorial



AURIX™ TC3xx EDSADC
Basic hands-on tutorial for resolver carrier cancelation

AP32445
v1.1

About this document

Scope and purpose

- This tutorial describes how to configure EDSADC for resolver support with an external carrier application and how to provide register settings for a quick trial

- This document is valid for AURIX™ family devices
  - TC38x AA-step
  - TC39x BA-step

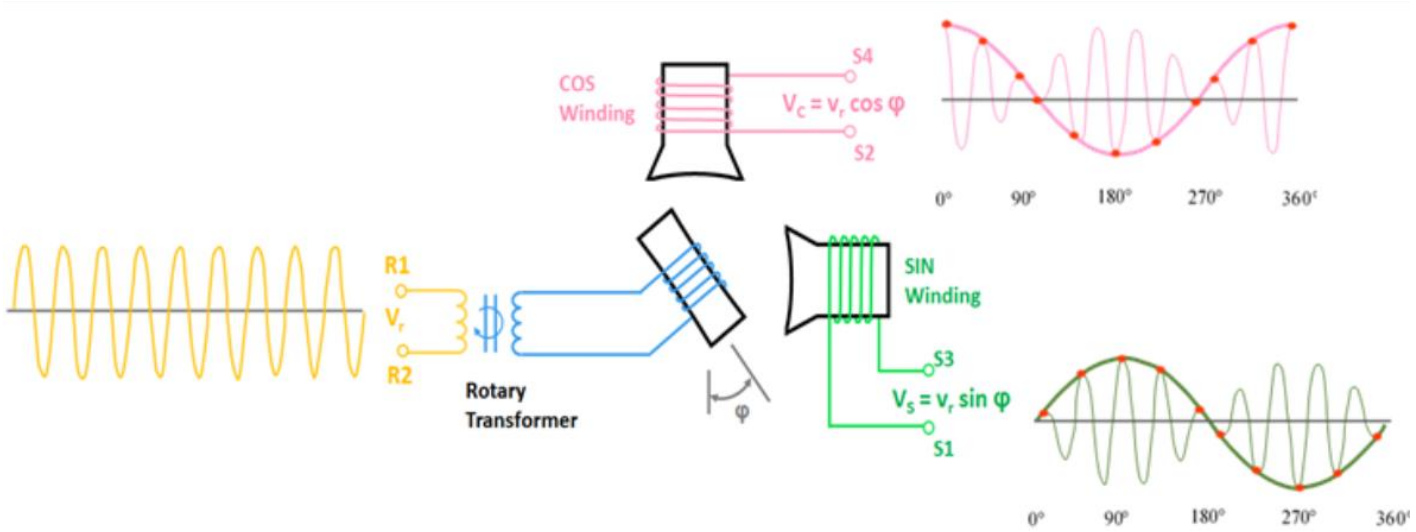
- Validated on TC399-B-Step

Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

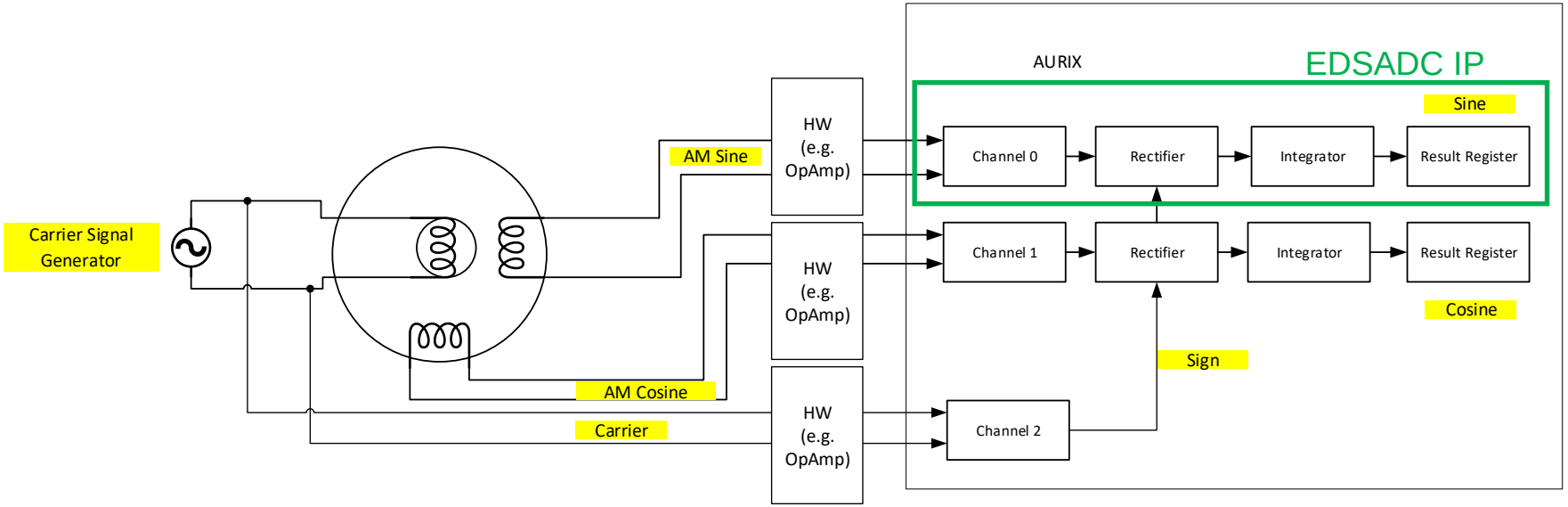
Agenda

Task and signal conditioning



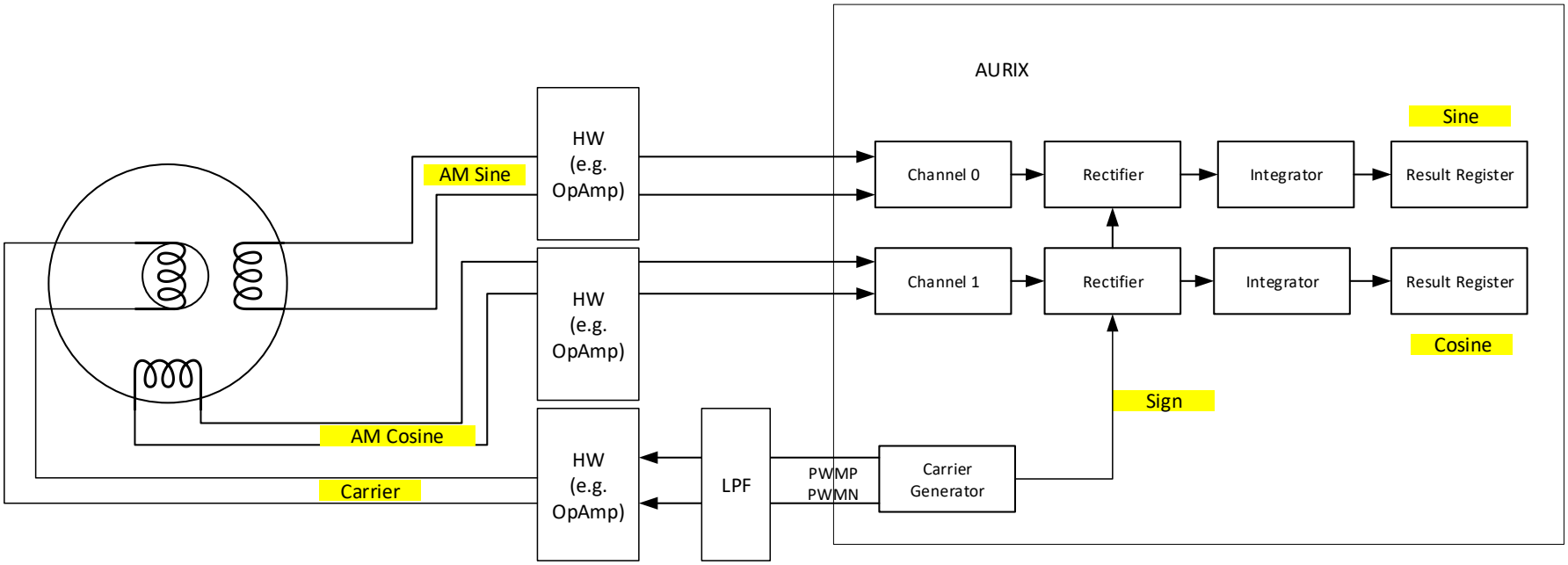
- Demodulate the resolver signal from AM modulated carrier signal
- The demodulated signal is used for motor position and speed estimation

External carrier generator use case



- The carrier signal is generated by external circuitry
- AURIX™ needs three DSADC channels to do the coherent carrier demodulation
- Channel 2 obtains only the sign information for rectification for the integrator

Internal carrier generator use case

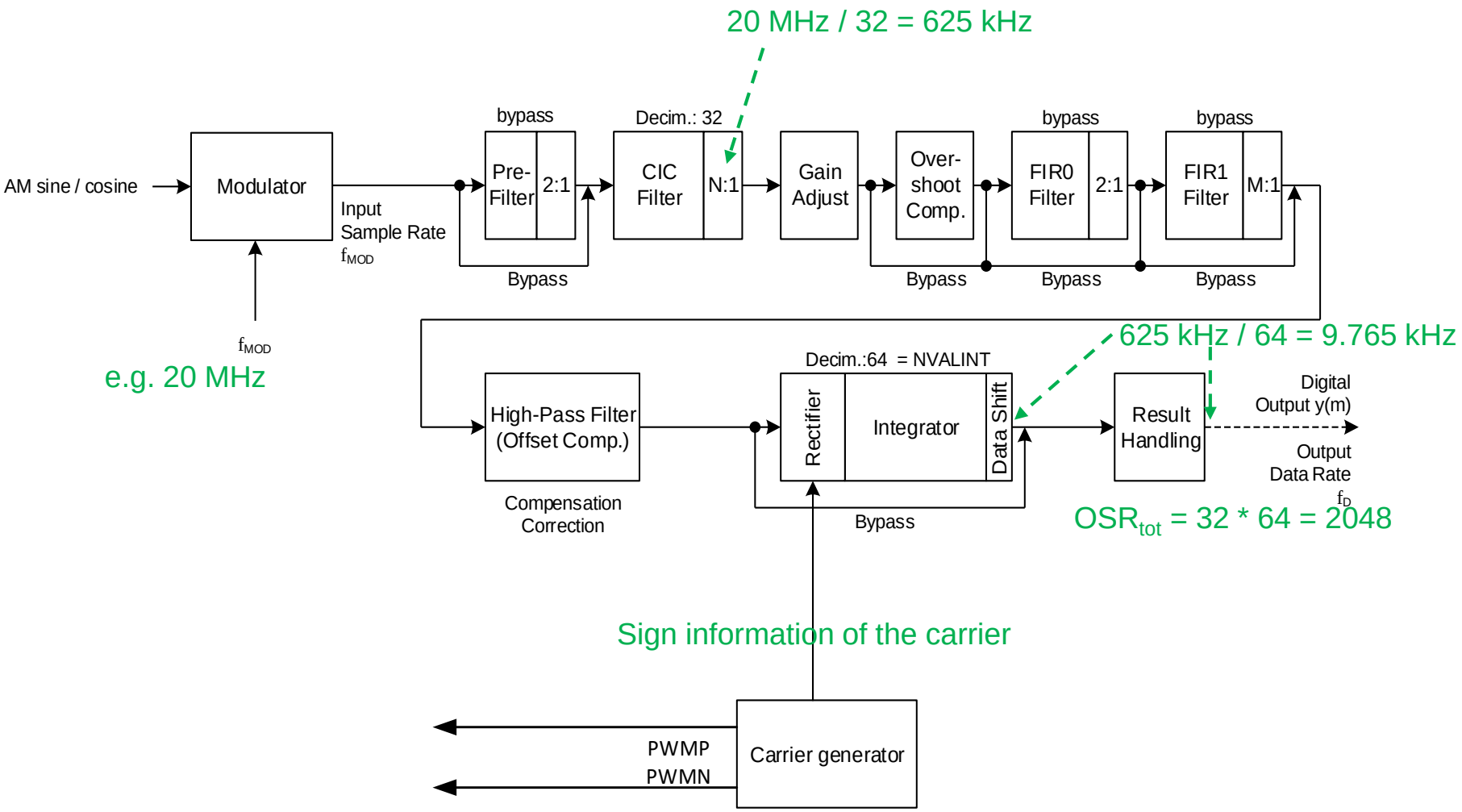


- The carrier signal is generated by internal circuitry and filtered by low pass filter (LPF) on PCB
- AURIX™ needs two DSADC channels to do the coherent carrier demodulation

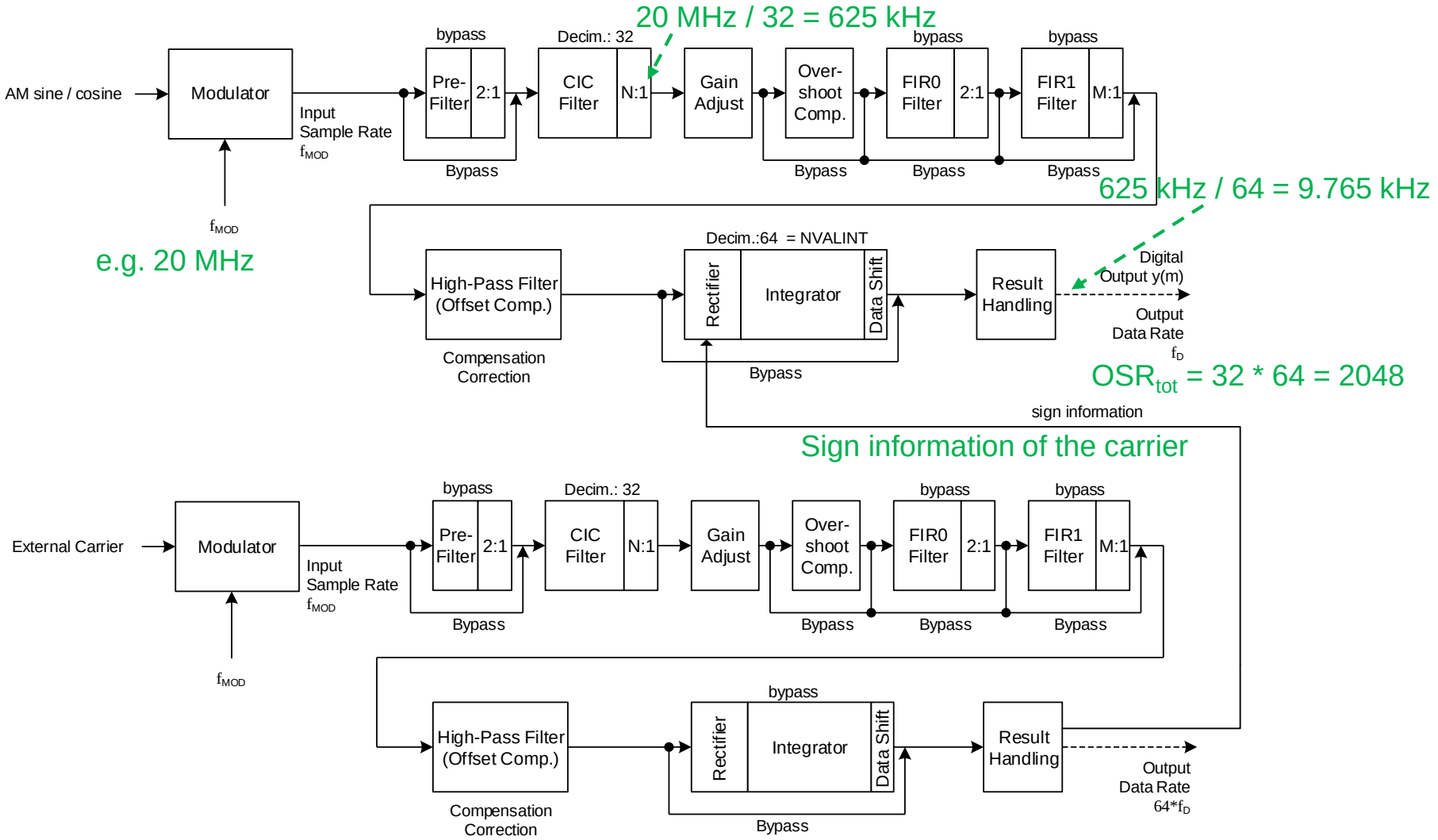
Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

EDSADC for internal carrier use case



EDSADC for external carrier use case



EDSADC important system conditions

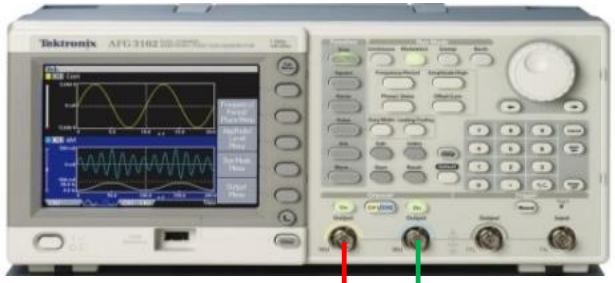
- To support an optimized carrier cancelation, following configurations are mandatory:
  - An even number of integration cycles has to be configured
  - The total oversampling rate (OSR_{tot}) costing of configured prefilter decimation, CIC decimation, FIR0/1 decimation, number of integration cycles has to fulfill the flowing condition:
  - The quotient (OSR_{tot}) between configured modulator clock (f_{mod}) and the carrier frequency ($f_{carrier}$) has to be an integer value
    → $OSR_{tot} = f_{mod} / f_{carrier}$
  - In case external carrier generator is used, OSR_{tot} must be guaranteed for the coherent demodulation

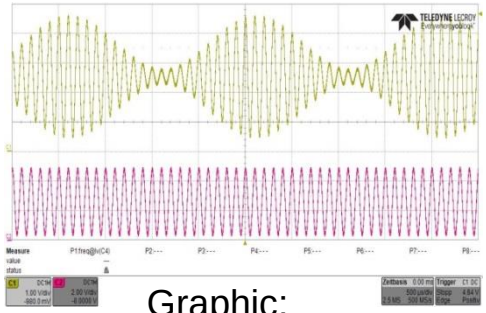
Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

Live session trial for external carrier case

Tektronix AFG3252
Dual Channel
Function Generator
(just an example → any signal generator can be used, which supports amplitude modulation)





Graphic:
rawdata[$]

Amplitude modulated Signal

Carrier 9.765 kHz

EDSADC0

EDSADC1

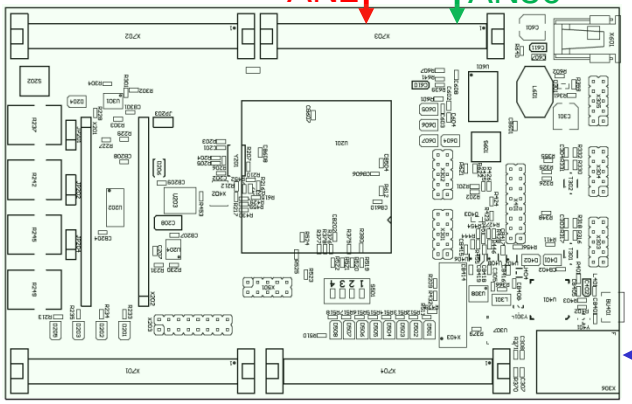
Debugger

DSADC Results

AURIX™ Triboard

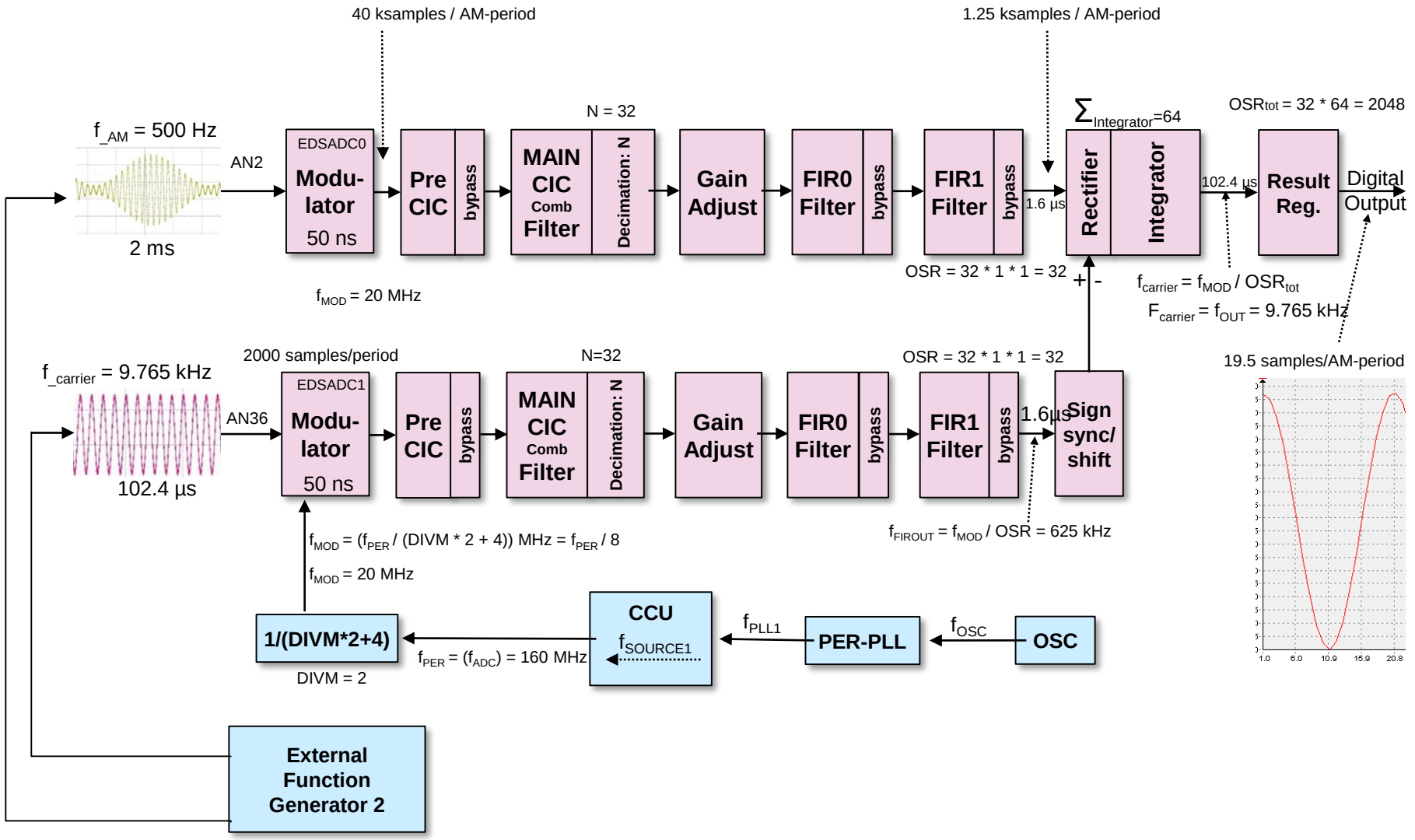
AN2

AN36



USB/DAS

TC3xx: EDSADC carrier cancellation
Test setup block diagram



Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

TC3xx: EDSADC carrier cancellation
Register settings

Name	Register settings:
EDSADC_GLOBCFG	0x80008000 ⇨ Sync. Mode
EDSADC_MODCFG0	0x88028008 ⇨ INCFGN = Vrefx, DIVM = 2
EDSADC_MODCFG1	0x88028008 ⇨ INCFGN = Vrefx, DIVM = 2
EDSADC_VCM0	0x00000004 ⇨ Vrefx = Varef / 2
EDSADC_VCM1	0x00000004 ⇨ Vrefx = Varef / 2
EDSADC_DICFG0	0x84308000 ⇨ ITRMODE: Integrator always active
EDSADC_DICFG1	0x84008000 ⇨ ITRMODE: Integrator bypassed
EDSADC_GAINCORR0	0x000F1194 ⇨ CICSHIFT = 15, GAINFACTOR = 1194
EDSADC_GAINCORR1	0x000F1194 ⇨ CICSHIFT = 15, GAINFACTOR = 1194
EDSADC_FCFGC0	0x001F001F ⇨ CIC Filter Decimation Factor = 32
EDSADC_FCFGC1	0x001F001F ⇨ CIC Filter Decimation Factor = 32
EDSADC_FCFGM0	0x80038000 ⇨ FIR0/1 off, Prefi off, calib off
EDSADC_FCFGM1	0x80038000 ⇨ FIR0/1 off, Prefi off, calib off
EDSADC_IWCTR0	0x3F000005 ⇨ 64 integration cycles
EDSADC_CGSYNC0	0x20000000 ⇨ SDNEG = 32
EDSADC_RECTCFG0	0x00000111 ⇨ Rectifier enabled, sign-source ch1
EDSADC_GLOBRC	0x00030003 ⇨ Enable modulator/demod 0 and 1

Important system considerations #1

GLOBRC
Global Run Control Register **(0088$_H$)** **Application Reset Value: 0000 0000$_H$**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0		M13RUN	M12RUN	M11RUN	M10RUN	M9RUN	M8RUN	M7RUN	M6RUN	M5RUN	M4RUN	M3RUN	M2RUN	M1RUN	M0RUN
r		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		CH13RUN	CH12RUN	CH11RUN	CH10RUN	CH9RUN	CH8RUN	CH7RUN	CH6RUN	CH5RUN	CH4RUN	CH3RUN	CH2RUN	CH1RUN	CH0RUN
r		rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw	rw

- Start the modulators and the demodulators always synchronously (never sequentially)
- (pseudo) code example:

```
DSADC_GLOBRC = 0x00030000;  // start the needed modulators at once
wait 20 µs  // modulator startup time
DSADC_GLOBRC = 0x00030003; // start the needed demodulators at once
```

The modulators and demodulators must only be started at the end of the initialization sequence!

Important system considerations #2

- The carrier generator must be enabled just before the demodulator to avoid any variation in the angle calculation between power cycles!

```
//enable Modulators
EDSADC_GLOBRC.U = 0x000F0000;
wait(200, 100);  // (at least 20 µs) for modulator startup time
// disable all interrupts
// enable CARRIER generator
EDSADC_CGCFG.U =0x00000073;
//enable Demodulators
EDSADC_GLOBRC.U = 0x000F000F;
// enable interrupts
```

Details how to calculate gain correction values

CICSHIFT = roundup (14 – ld(2 * AFS / (N^3 * 4 * FM)))
CICSHIFT = roundup (14 – ld(2 * 25000 / (10^3 * 4 * 0.7)))
CICSHIFT = roundup (14 – ld(50/2.8)) = roundup(14 – log17.9/log2)
CICSHIFT = roundup (14 – 4.158) = roundup(9.84) = 10

AFS = calibrated full-scale value
= 25 000 (after reset)
N = selected CIC decimation factor
FM = modulator gain factor
– on-chip modulator: FM = 0.7

GainCorrectionFactor = (2 * AFS / (N^3 * 4 * FM)) * 2^(CICSHIFT-14))
GainCorrectionFactor = (2 * 25000 / (10^3 * 4 * 0.7)) * 2^(10 - 14)
GainCorrectionFactor = 50 / (4 * 0.7) * 2^-4
GainCorrectionFactor = 1.11607
GAINFACTOR = GainCorrectionFactor * 4096
GAINFACTOR = 1.1160714 * 4096
GAINFACTOR = 4571 = $11DB_H$ (bitfield value)

In this example CIC decimation factor:
N = 10

Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

Programming example

- The programming example is written for TC38x AA-step / TC39x BA-step. With minor adjustments it can also be used for other devices of the TC3xx family

- Important notices:

  - Infineon Technologies AG (Infineon) provides this programming example for exclusive use with Infineon microcontroller products. This programming example can be freely distributed within development tools that support such microcontroller products

  - This software is provided "as is". No warranties, whether express, implied or statutory, including, but not limited to, implied warranties of merchantability and fitness for a particular purpose apply to this software. In no event shall Infineon be liable for any special, incidental or consequential damages, for any reason whatsoever

Programming example
Converter control and enable EDSADC

```
//----- Clock Control Reg: enable module clock, sleep mode request enabled

    IfxScuWdt_clearCpuEndinit(l_EndInitPW);
    CONVCTRL_CLC.U = 0x00000000;                // enable module clocks
    while ((CONVCTRL_CLC.U & 0x00000002) == 2); // wait until module is enabled

    //----- Bus Peripheral Interface: reset kernel0/1
    CONVCTRL_KRST0.B.RST = 0x1;         // reset kernel0
    CONVCTRL_KRST1.B.RST = 0x1;         // reset kernel1
    while (!CONVCTRL_KRST0.B.RSTSTAT); // wait until reset is performed
    CONVCTRL_KRSTCLR.B.CLR = 0x1;       // clear kernel reset status bit

    CONVCTRL_CCCTRL.B.TC = 0xB;         // Access to Converter Control registers is enabled

    //----- set ADC phase synchronization

    CONVCTRL_PHSCFG.U = 0x00008007;     //phase synchronization signal is generated at fadc / 8
    IfxScuWdt_setCpuEndinit(l_EndInitPW);


    //----- enable DSADC ----------------------------------------------------------------------
------

    IfxScuWdt_clearCpuEndinit(l_EndInitPW);
    EDSADC_CLC.U = 0;                           // load clock control register
    while ((EDSADC_CLC.U & 0x00000002) == 2);    // wait until module is enabled
    // reset module
    EDSADC_KRST0.U = 1;
    EDSADC_KRST1.U = 1;
    while (EDSADC_KRST0.U != 0x2);
    // clear the set reset flag
    EDSADC_KRSTCLR.U = 1;
    IfxScuWdt_setCpuEndinit(l_EndInitPW);
```

Programming example
EDSADC initialization

```
//----- initialize DSADC  for CIC32 and INT 64-----------------------------------------------------------------

    EDSADC_GLOBCFG.U  = 0x00000000;  // synchronized clock gen., no supervision channel

    //----- Modulator Demodulator
    EDSADC_MODCFG0.U  = 0x88028008;  // INCFGN=Vrefxff, DIVM=2
    EDSADC_MODCFG1.U  = 0x88028008;  // INCFGN=Vrefx
    EDSADC_VCM0.U     = 0x00000004;  // Vrefx=Varef/2
    EDSADC_VCM1.U     = 0x00000004;  // Vrefx=Varef/2
    EDSADC_DICFG0.U   = 0x84308000;  // ITRMODE: integrator always active
    EDSADC_DICFG1.U   = 0x84008000;  // ITRMODE: integrator bypassed

    //----- Gain
    EDSADC_GAINCORR0.U = 0x000F1194;
    EDSADC_GAINCORR1.U = 0x000F1194;

    //----- Filter(CIC, FIR0, FIR1)
    EDSADC_FCFGC0.U   = 0x001F001F;  // CIC Filter Decimation Factor = 32
    EDSADC_FCFGC1.U   = 0x001F001F;  // CIC Filter Decimation Factor = 32

    //----- Integrator Rectifier Carrier Synchronization
    EDSADC_IWCTR0.U   = 0x3F000005;  // 64 integration cycles
    EDSADC_CGSYNC0.U =  0x20000000;  // SDNEG=32
    EDSADC_RECTCFG0.U = 0x00000111;  // Rectifier enabled, sign source ch1

    //----- enable modulator/demodulator
    EDSADC_GLOBRC.U   = 0x00030000;  // Enable modulator 0 and 1
    wait(1000, 100); //DSADC wakeup time ~28µs
    EDSADC_GLOBRC.U   = 0x00030003;  // Enable demod 0 and 1

    //----- Filter (FIR0, FIR1), enable service request, start calibration
    EDSADC_FCFGM0.U   = 0x80038000;  // FIR0/1 off, Deci=2, Pref. off, No calibration
    EDSADC_FCFGM1.U   = 0x80038000;  // FIR0/1 off, Deci=2, Pref. off, No calibration
```

Programming example
Conversion loop

```
//----- variables ------------------------------------------------------------
   sint16   rawdata_amplitude[1000];
   sint16   rawdata_carrier[1000];
   uint32 ii = 0;
//----------------------------------------------------------------------------
```

```
for (ii = 0; ii < 1000; ii += 1)
{
    //----- store carrier signal
      while ((SRC_DSADCSRM0.U & 0x01000000) == 0);//SRC_EDSADCSRM0.SRR[24]: Service Request Flag, DSADC SRMm Service
Request
      conv16     = (sint16) EDSADC_RESM0.U;             // Result Reg0 Main Filter
      rawdata_carrier[ii] = conv16;
      SRC_DSADCSRM0.U = 0x02000000;               //SRC_EDSADCSRM0.CLRR[25]: Request Clear Bit
    //----- store amplitude modulation signal
      while ((SRC_DSADCSRM1.U & 0x01000000) == 0);//SRC_EDSADCSRM0.SRR[24]: Service Request Flag, DSADC SRMm Service
Request
      conv16     = (sint16) EDSADC_RESM1.U;             // Result Reg2 Main Filter
      rawdata_amplitude[ii] = conv16;
      SRC_DSADCSRM1.U = 0x02000000;               //SRC_EDSADCSRM2.CLRR[25]: Request Clear Bit

      if(ii >= 999) ii = 0;  //loop cycle
      }
```

Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

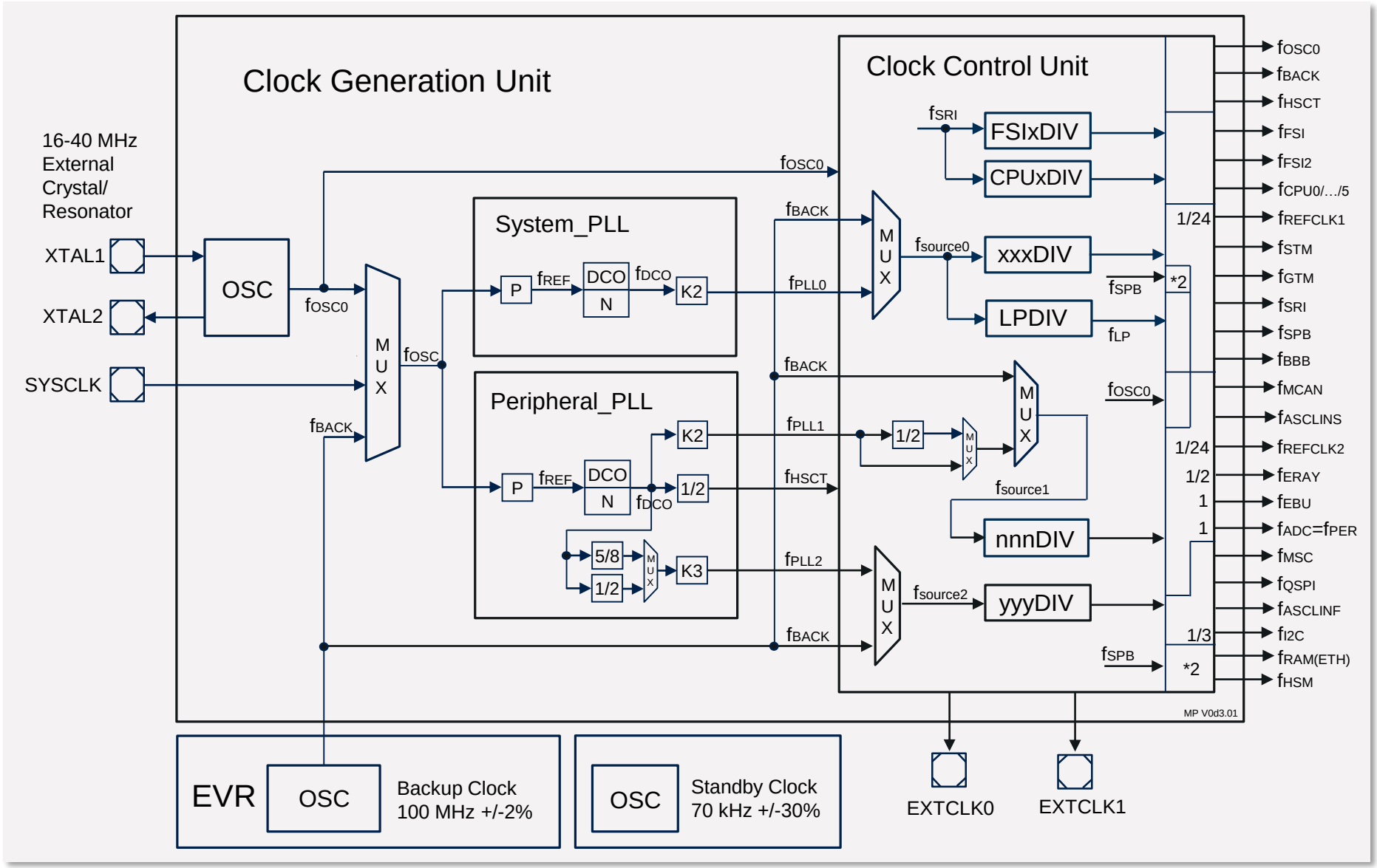
Summary

- AURIX™ EDSADC resolver support is based on coherent demodulation technique:
  - The quotient (OSR_{tot}) between configured modulator clock (f_{mod}) and the carrier frequency ($f_{carrier}$) has to be an integer value
    → $OSR_{tot} = f_{mod} / f_{carrier}$
- Filter chain for resolver application is CIC filter and Integrator (FIR filters are not needed)
- Consider group and hardware delay in systems and use SDPOS and SDNEG for compensation (separate tutorial)
- Avoid any interrupts during modulator enabling and carrier generator enabling
- For more details on register settings on internal carrier case see also the Appnote Tutorial AP32459

Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

Clock generation
Typical block diagram



Agenda

1. Resolver system overview
2. EDSADC filter chain configurations
3. Hands-on setup block diagram
4. Register settings: Debugger screenshot
5. Code example
6. Summary
7. Clock generation: Typical block diagram
8. Acronyms, references, revision history

Acronyms, references, revision history

Acronyms

Term	Definition
AM	Amplitude modulated
CCU	Clock Control Unit
CIC	Cascaded Integrator Comb(filter)
DICFG	Demodulator Input Configuration Register
EDSADC	Enhanced Delta-Sigma Analog-to-Digital Converter
FCFGC	Filter Configuration Register CIC Filter
FCFGM	Filter Configuration Register Main
FIR	Finite Impulse Response(filter)
GAINCORR	Gain Correction Register
GLOBCFG	Global Configuration Register
GLOBRC	Global Run Control Register
GSYNCC	Carrier Generator Synchronization Register
HW	Hard Ware
IWCTR	Integration Window Control Register
MODCFG	Modulator Configuration Register
OpAmp	Operational Amplifier

Acronyms, references, revision history

Term	Definition
OSC	Oscillator
OSR	Oversampling Ratio
PER-PLL	Peripheral PLL
PCB	Printed Circuit Board
RECTCFG	Rectification Configuration Register
VCM	Common Mode Voltage Register

References

- [1] AURIX_TC3xx_Safety_Manual_V1_12
- [2] AURIXTC3XX_um_part2_V2.0

Revision history

Revision	Description of change
V1.0	Initial version
V1.1	Update of typo for the EDSADC_GLOBRC register fields. First the modulators must be enabled an afterwards the demodulators

Part of your life. Part of tomorrow.

Trademarks
All referenced product or service names and trademarks are the property of their respective owners.

Published by
Infineon Technologies AG
81726 Munich, Germany

Do you have a question about this document?
Email: erratum@infineon.com