AN4413
Application note

SPC57xx ADC device

Introduction

The aim of this document is to clarify the usage and features compared to the SPC57xx ‘s ADC, in order to help the user to find the best configuration to optimize the device use, in term of quality/precision of the sampled signals.

Document is intended as a guideline for the SARADC and SDADC features.

Contents

List of tables

List of figures

1 SAR ADC characteristics

The SAR ADC device consists of two kind of ADC: a fast ADC (SAR ADC) and a slow ADC (SARB ADC).

All analog input pins routed to the SAR ADCB and the other fast SAR ADC is multiplexed with a dual analog input switch pad cell. Simultaneous sampling of the two ADCs on a same analog input is not allowed.

In the *Figure 1* the block diagram of SAR ADC:

Figure 1. SAR ADC block architecture

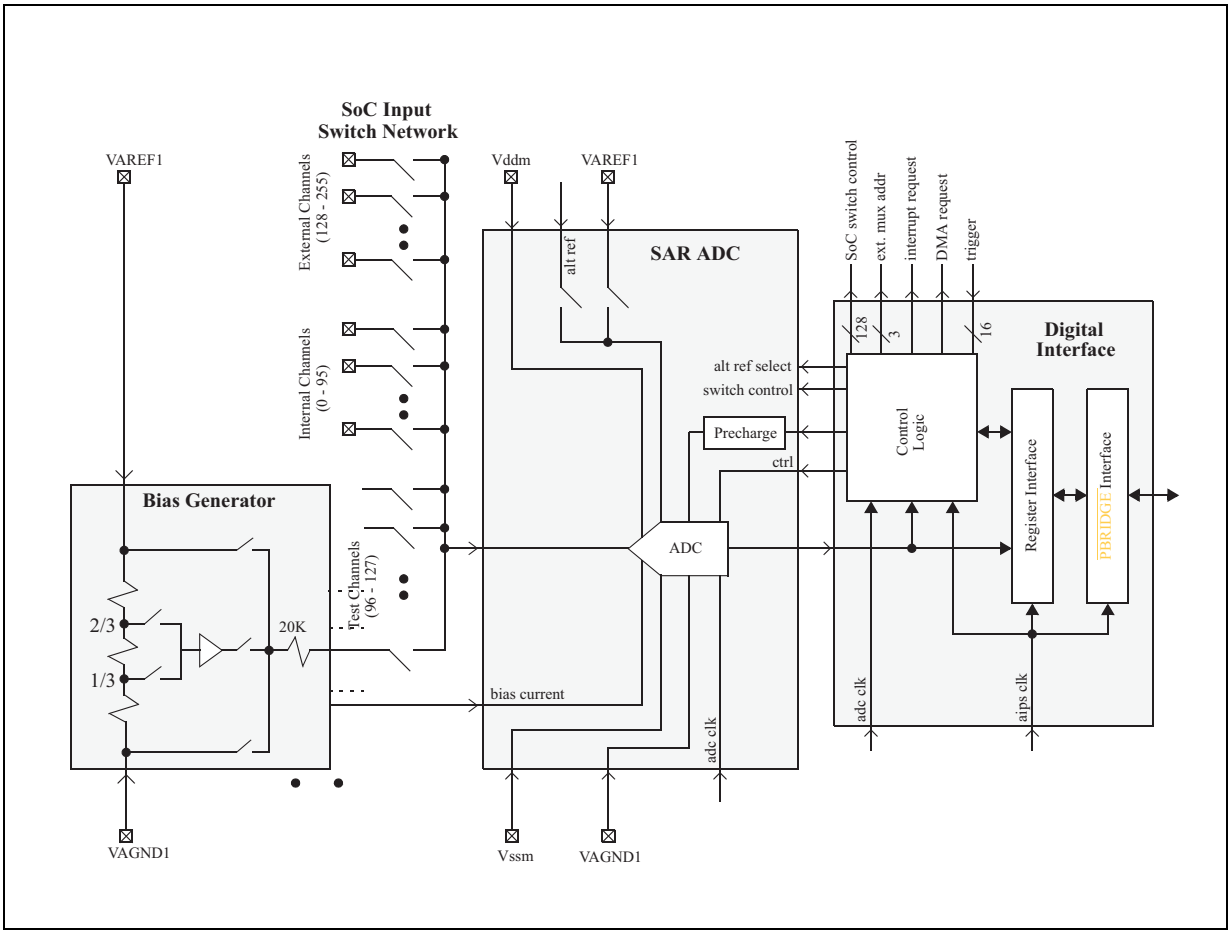
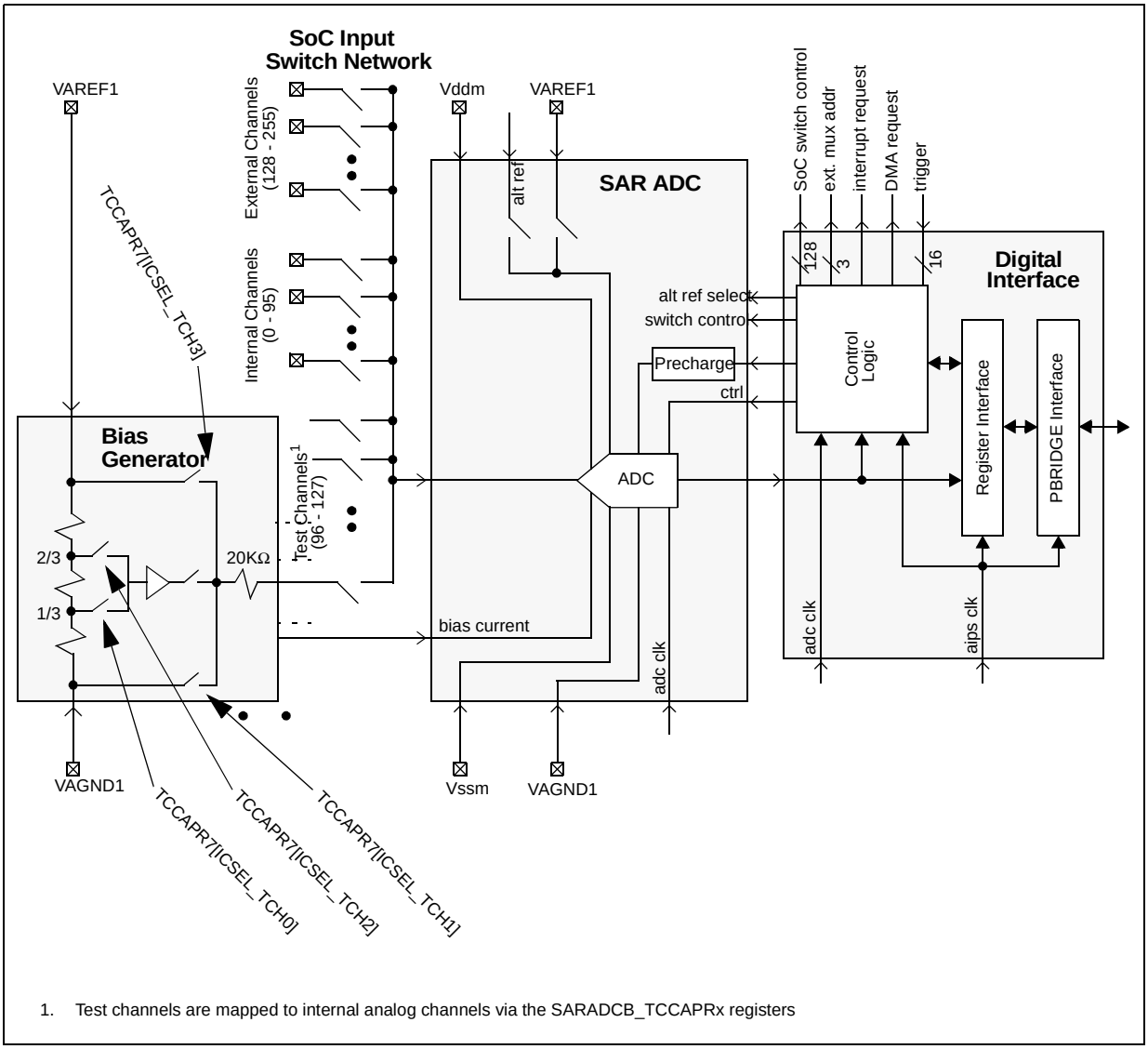


Figure 2. SARADC block diagram



1. Test channels are mapped to internal analog channels via the SARADCB_TCCAPRx registers

1.1 Features

- 12 bit resolution
- Reference voltage: from 2.0V to 6.0V
- Supply voltage: from 3.5V to 5.5V
- Maximum clock frequency: 14.4MHz
- Programmable sampling time
- Software controlled Power-down

Please refer to Datasheet for detailed features.

1.2 Considerations on the input impedance of the signal source

To preserve the accuracy of the ADC, it is necessary that analog input pins have low AC impedance. Placing a capacitor with good high frequency characteristics at the input pin of the device, can be effective: the capacitor should be as large as possible, ideally infinite.

This capacitor contributes to attenuating the noise present on the input pin; furthermore, its sources charge during the sampling phase, when the analog signal source is a high impedance source.

The impedance relative to the signal source can limit the ADC's sample rate. Furthermore a current limiter resistance and an RC filter are often necessary to minimize the current request and to attenuate the noise present on the input pin. This external network can generate accuracy problems for the ADC converter and for this reason it is important to invest time in reaching the right adaptation.

Figure 3. SAR ADC input equivalent circuit

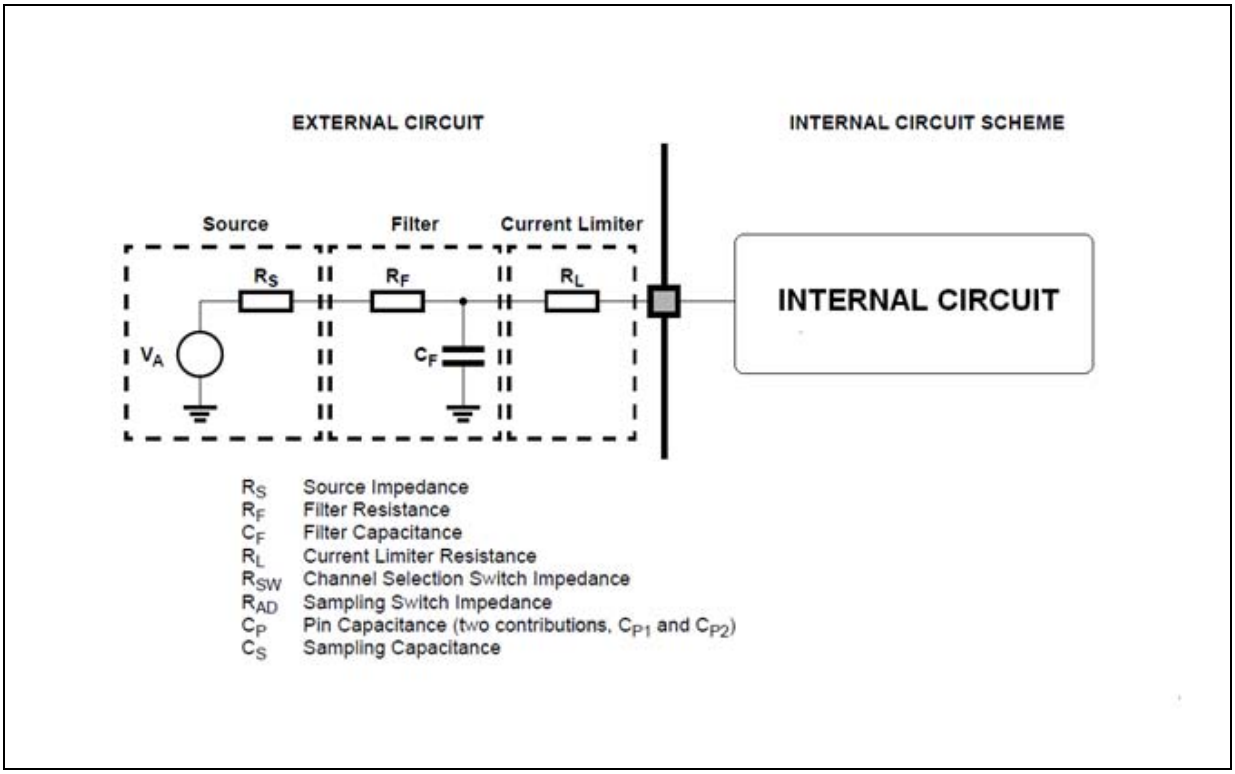


Figure 4. Input equivalent circuit (Fast SARn channels)

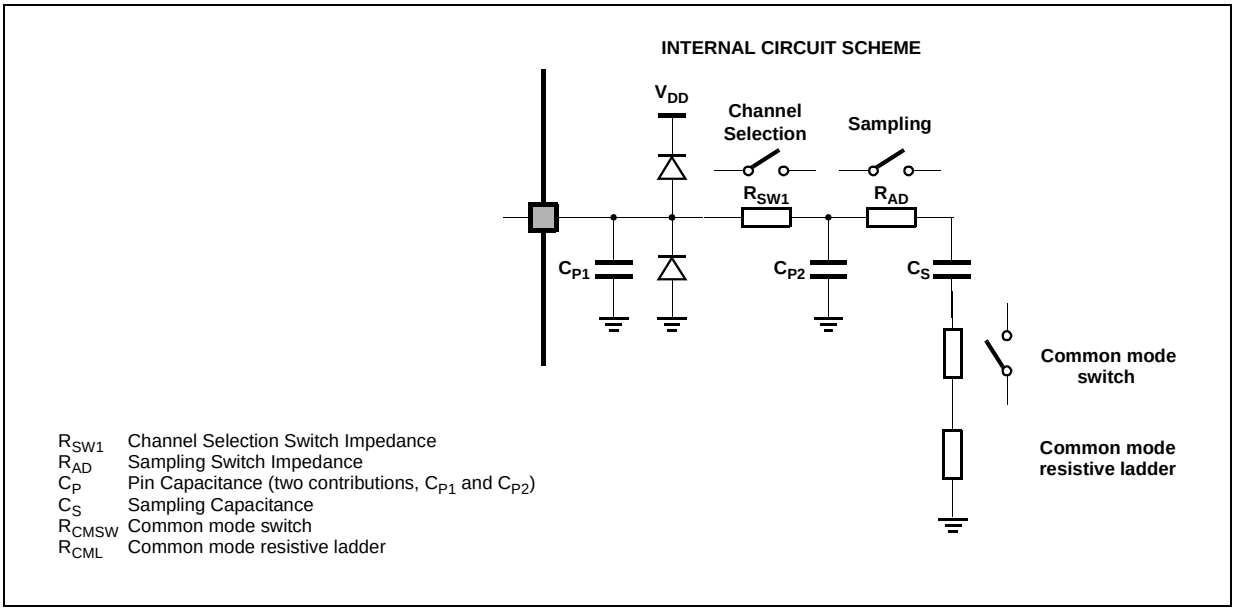
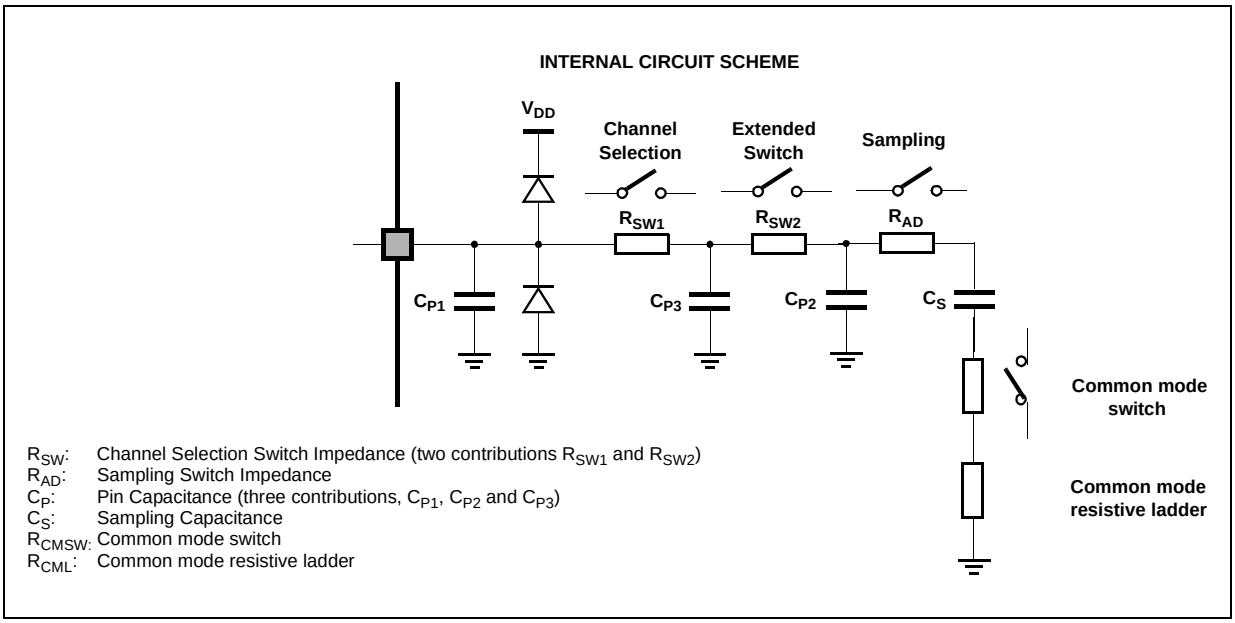


Figure 5. Input equivalent circuit (SARB channels)



The sampling capacitor Cs of the SAR, can be seen as a switching current sink element. The sampling capacitor switching at the conversion rate of the input channel can be seen as a resistive path to ground. For instance assuming a conversion rate of 400 ksps, with Cs equal to 6 pF a resistance of 417 kΩ (Req = 1/(fc.Cs)) is obtained.

To minimize the error induced by the voltage partitioning between this resistance (sampled voltage on CS) and the sum of RS + RF, the external circuit must be designed to respect the following relation.

$$V_A \cdot \frac{R_S + R_F}{R_{eq}} < \frac{1}{2}LSB$$

which leads to follow this expression:

$$R_S + R_F < \frac{1}{(8192 \cdot f_C \cdot C_S)}$$

The formula above provides a constraint for the external network design, in particular on the resistive path.

The current limiting resistance value impacts the sampling time of the SAR. In order to run the SAR at the highest sampling speed, R_L should respect the following constraint:

$$R_L < 350 Ohm$$

Of course, RL shall be sized also according to the current limitation constraints, in combination with RS (source impedance) and R_F (filter resistance), due to some consideration on Nyquist theorem, and transfer charge, then a constraints on Cs is:

$$C_F > 8192 \cdot C_S$$

1.3 Conversion timings

In order to support different loadings and switching times four different Conversion Timing Registers are present (CTR0–3). Each conversion timing register contains PRECHG, INPSAMP bitfields to program the required duration for precharging and sampling phases. The selection of these registers for each channel is done by the SAMPSEL bitfield of the corresponding channel data register.

Bitfields PRECHG, INPSAMP are used to define the total conversion duration (t_{conv}) and in particular the partition among precharge duration (t_{prechg}), sampling phase duration (t_{sample}) and evaluation phase duration (t_{eval}).

The precharging phase duration is given by

$$t_{prechg} = PRECHG * t_{ck}$$

where PRECHG ≥ 2 (hardware requirement) and PCE bit of channel data register is ‘1’. In case the value of the PRECHG bit field is less than 2, it is automatically set to 2 inside the formula.

If the PCE bit is ‘0’, the precharging phase is skipped and the conversion starts with a sampling phase directly.

The sampling phase duration is given by the following equation:

$$t_{sample} = INPSAMP * t_{ck}$$

where INPSAMP ≥ 4 (hardware requirement). In case the value of the INPSAMP bit field is less than 5, it is automatically set to 5 inside the SARADC.

The total evaluation phase duration is given by the following equations:

$$t_{eval} = 25 * t_{ck}$$

$$t_{eval} = 21 * t_{ck} \text{ (for 10-bit conversion)}$$

The total conversion duration is (not including external multiplexing) given by the following:

$$t_{conv} = t_{prechg} + t_{sample} + t_{eval}$$

The timings refer to the unit t_{ck} refers to reciprocal of f_{ck}, where f_{ck} = SARADC peripheral clock.

1.4 SARADC: constrain and timing

The integration of the ADC 10-bit module in the SARADC device leads to some further constraints to guarantee the performance indicated in the datasheet/electrical characteristics. In particular sampling time and conversion times must respect the minimum values, which are:

- 10 bit:
  - Sampling time must be equal or greater than 555 ns
  - Evaluation time must be equal or greater than 1454 ns
- 12 bit slow (SARADC_B):
  - Precharge time must be equal or greater than 540 ns
  - Sampling time must be equal or greater than 1500 ns
  - Evaluation time must be equal or greater than 1712 ns
- 12 bit fast (SARADCx):
  - Precharge time must be equal or greater than 270 ns
  - Sampling time must be equal or greater than 750 ns
  - Evaluation time must be equal or greater than 1712 ns

Table 1. 10 bit SARADC timing

Clock (MHz)	TCK (µs)	PRECHG	INPSAMP 10 bit (µs)	tPRECHG (µs)	tSAMPLE 10 bit (µs)	tEVAL 10 bit (µs)	tCONV 10 bit (µs)
2	0.500	2	4	1.000	2.000	10.500	13.500
4	0.250	2	4	0.500	1.000	5.250	6.750
6	0.167	2	4	0.333	0.667	3.500	4.500
8	0.125	3	5	0.375	0.625	2.625	3.625
10	0.100	3	6	0.300	0.600	2.100	3.000
12	0.083	4	7	0.333	0.583	1.750	2.667

Table 1. 10 bit SARADC timing (continued)

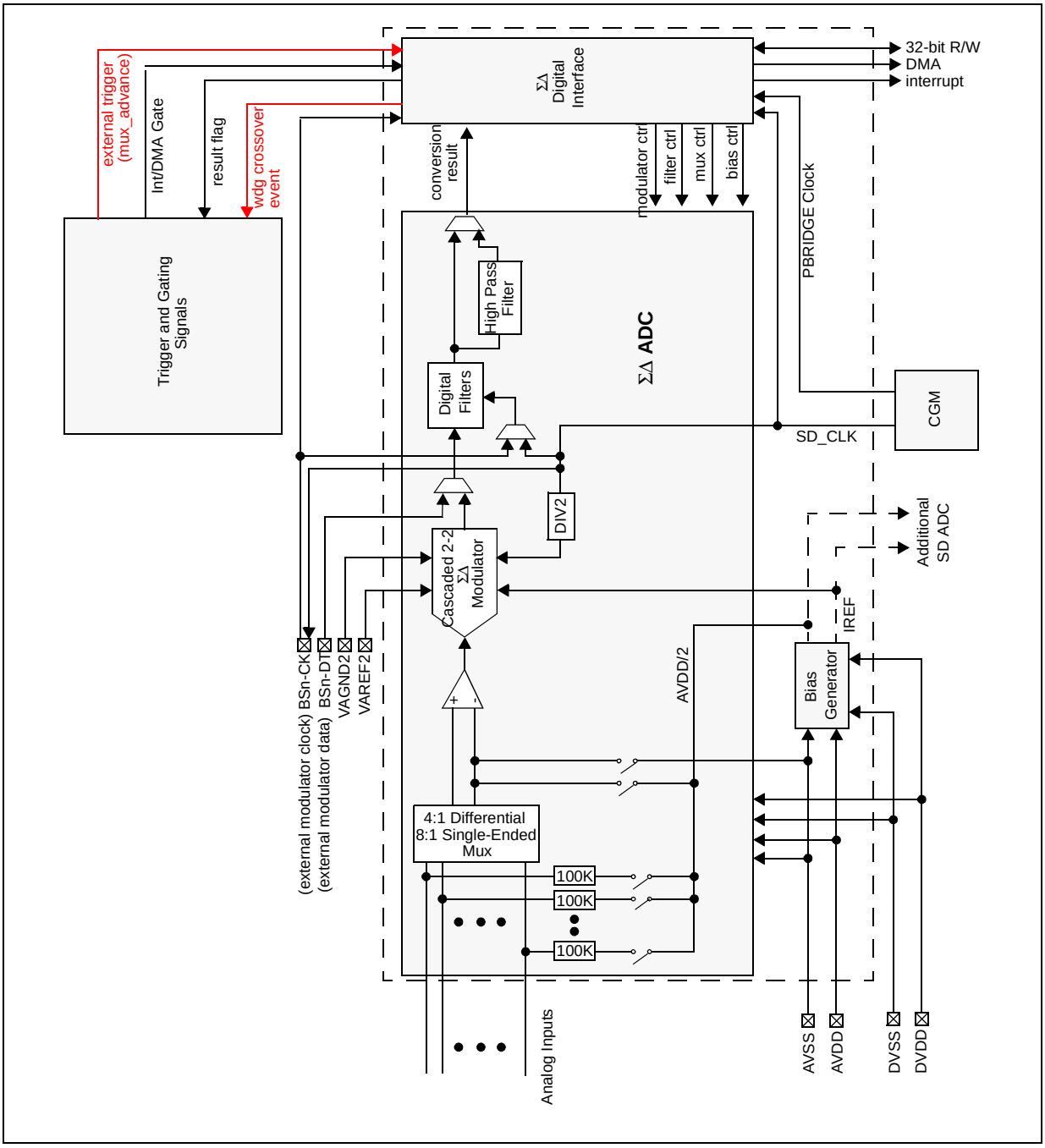
Clock (MHz)	TCK (µs)	PRECHG	INPSAMP 10 bit (µs)	tPRECHG (µs)	tSAMPLE 10 bit (µs)	tEVAL 10 bit (µs)	tCONV 10 bit (µs)
14	0.071	4	8	0.286	0.571	1.500	2.357
14.6	0.068	4	9	0.274	0.616	1.438	2.329

2 SD ADC characteristics

The Sigma-Delta Analog-to-Digital Converter (SDADC) digital interface block controls the on-chip SDADC feature. It provides an accurate conversion data and a conversion status for a wide range of applications.

In the figure below the block diagram of the SD ADC:

Figure 6. SDADC block architecture



2.1 Features

- Conversion rate (Fs): up to 300 Ksps
- OSR = selectable from 24 to 256
- Passband = 0.33*Fs
- Programmable gain
- Differential/single ended operation
- Rail to rail (0 to 5 V) input common mode range
- High input impedance
- Very low current from references
- Provision for external modulator
- Selectable OSR/decimation rate
- Provision for gain/offset calibration

Please refer to datasheet for detailed features.

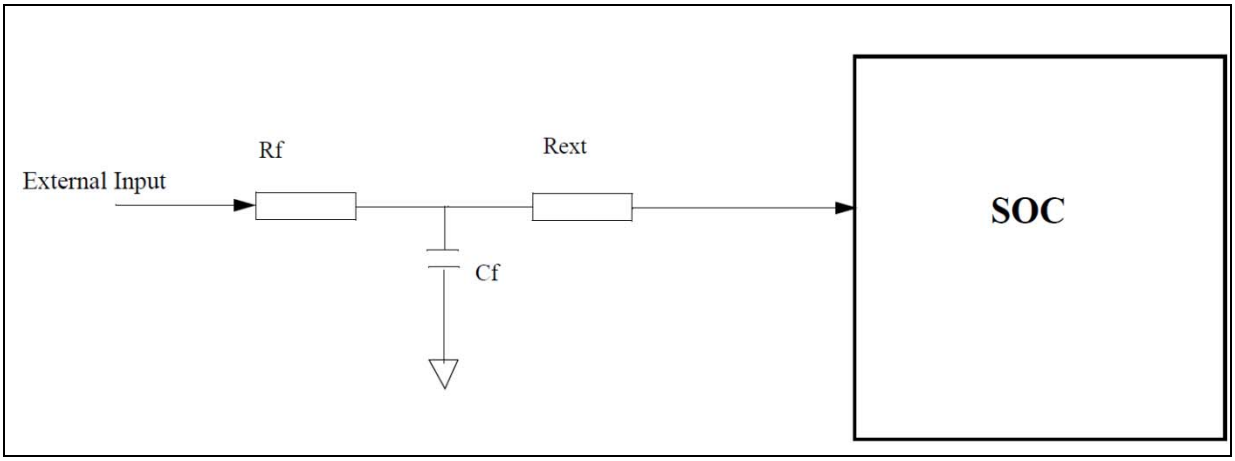
2.2 Considerations on the input impedance of the signal source

To preserve the accuracy of the ADC, it is necessary that analog input pins have low AC impedance. Placing a capacitor with good high frequency characteristics at the input pin of the device, can be effective: the capacitor should be as large as possible, ideally infinite.

This capacitor contributes to attenuate the noise present on the input pin; furthermore, its sources charge during the sampling phase, when the analog signal source is a high impedance source.

The impedance relative to the signal source can limit the ADC's sample rate. Furthermore an RC filter is often necessary to minimize the current request and to attenuate the noise present on the input pin. This external network can generate accuracy problems for the ADC converter and for this reason it is important to invest time in reaching the right adaptation.

Figure 7. SDADC input equivalent circuit



The recommend values are:

- R_{est} < 1 kΩ
- C_f =1n to 10 nF
- R_f = max 1 kΩ

With this value the cutoff frequency is:

$$f_{cutoff} = \frac{1}{2\pi \cdot R_f \cdot C_f}$$

for

- C_f = 10 nF
- R_f = 1 kOhm
- f_{cutoff} ≈ 15 kHz

2.3 SDADC and DMA

When the first DMA request is being served (generated on FIFO Full condition) and on completion of this request then the SDADC will be disabled.

In order to prevent other FIFO FULL condition, it is necessary to disable the SDADC before disabling the DMA.

This means, it's necessary to disable the SDADC (and DMA transfer is enabled), in order:

- To stop the operation completely
- To change the channel configuration and re-initiate

In both these conditions, in order to have the operation safe it’s necessary to clear the DMA[EN] bit along with MCR[EN].

2.4 SDADC conversion timing

There are some registers and definitions to take in account for the SDADC conversion time:

- OSDR: the Output Settling Delay Register (OSD Field) provides a delay value to qualify the converted output data. The OSD field defines the delay to qualify the conversion data stored in the Converted Data Register (CDR). Whenever the SDADC block is reset in order to start the conversion from a fresh state, an internal timer is loaded with the OSD start value. The counter counts down with the output clock f_d until it reaches ‘0’ and then it generates a flag which qualifies the converted data.
- f_s is the input sampling clock frequency
- f_d is the output clock

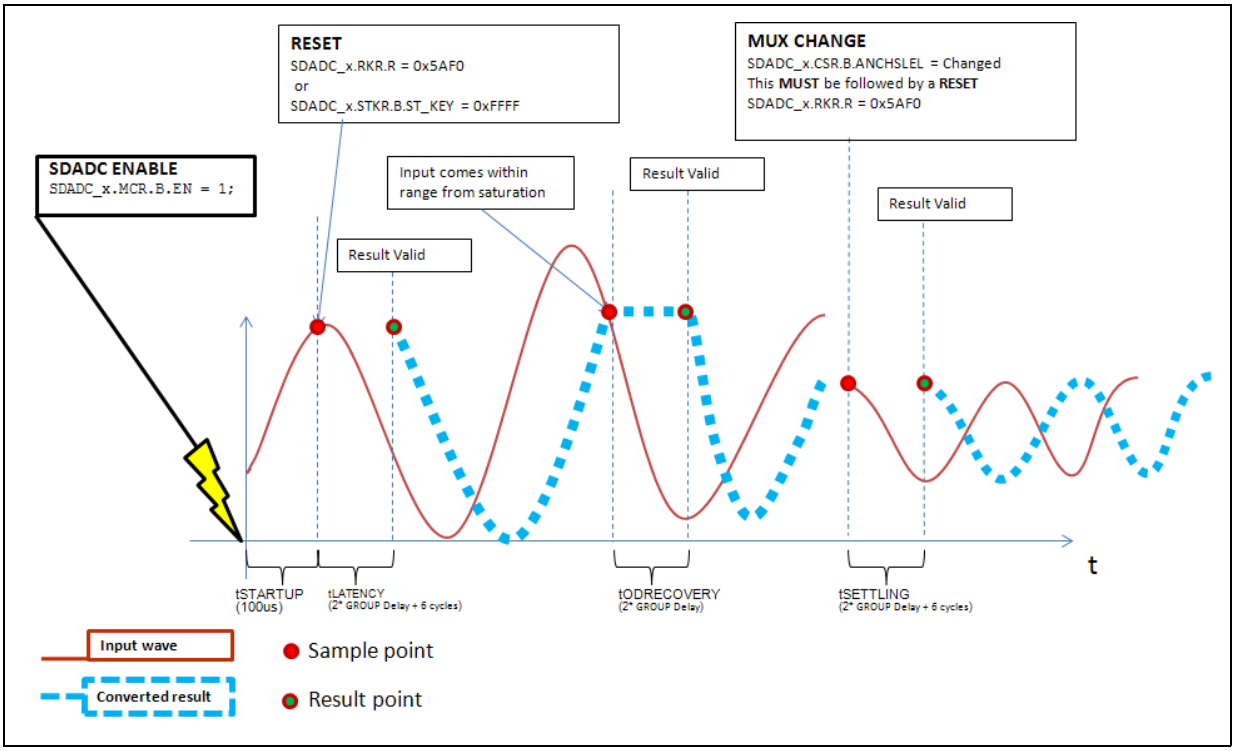
$$f_d = \frac{f_S}{2 \times OSR}$$

- δ_{GROUP} is the Groupdelay. The Groupdelay depend on OSR value
- tSTARTUP is the start-up time from the power down state. The ADC is “ready” EN is pulled to ‘1’. After tSTARTUP time it’s possible to apply a RESET and start a

conversion. The real data will come after the RESET is applied and you wait for “2*Groupdelay + 6 cycles” (6 cycles to RESET the application)

- tLATENCY = amount of time that passes from the converter capturing the analog signal until the digital output result is ready; tLATENCY = 2*Groupdelay + 6 cycles (6 cycles for RESET application)
- tSETTLING = settling time after the mux change. To mux change its need to apply a RESET. tSETTLING = 2*Groupdelay + 6 cycles (6 cycles to RESET the application)
- tODRECOVERY = overdrive recovery time, this means the time after input comes within range from saturation; tODRECOVERY = 2*Groupdelay.

Figure 8. Conversion delay



Just for example:

- SDADC_x.MCR.B.PDR = 0 (this means OSR = 24)
- Fclkout = Fclkin / (2*OSR) = Fclkinby2 / OSR
- Fclkinby2 = Fclkin / 2
- Fclkinby2 = Fclkout * OSR
- Tclkinby2 = Tclkout / OSR
- Groupdelay(OSR=24) = 187.5 * Tclkinby2 = (187.5 * Tclkout) / OSR = 187.5 / 24 ≈ 8 Tclkout
- tSETTLING = 2*Groupdelay + 6 cycles = 2*8 + 6 = 22
- Reset pulse width + Tprog + Tcapt ≈ 1
- SDADC_x.OSDR.B.OSD = 22 + 1 = 23

Since this calculation is valid for all OSR's values, then the minimum value of the OSD is 23.

2.5 Data conversion step

To acquire a data from the SDADC, the following sequence is required:

1. Enable the SDADC by asserting MCR[EN]
2. Configure MCR to select the required mode, polarity, common mode voltage, input gain, and decimation rate
3. Enable high-pass filter is required
4. Select the required analog channel for data conversion. It is possible to select the bias for each channel for AC coupling applications.
5. Configure the OSD delay according to the SDADC required startup time or latency from the reset exit
6. Generate a reset event writing 0x5AF0 in RESET_KEY field of the RKR register
7. Wait till FIFO empty flag DFEF of SFR register is clear
8. Read data by CDATA of CDR register

If you need to change of channel then:

1. Select the required analog channel for the data conversion. It is possible to select the bias for each channel for AC coupling applications
2. Generate a reset event writing 0x5AF0 in RESET_KEY field of the RKR register
3. Wait till FIFO empty flag DFEF of SFR register is clear
4. Read data by CDATA of CDR register

2.6 Data conversion timing

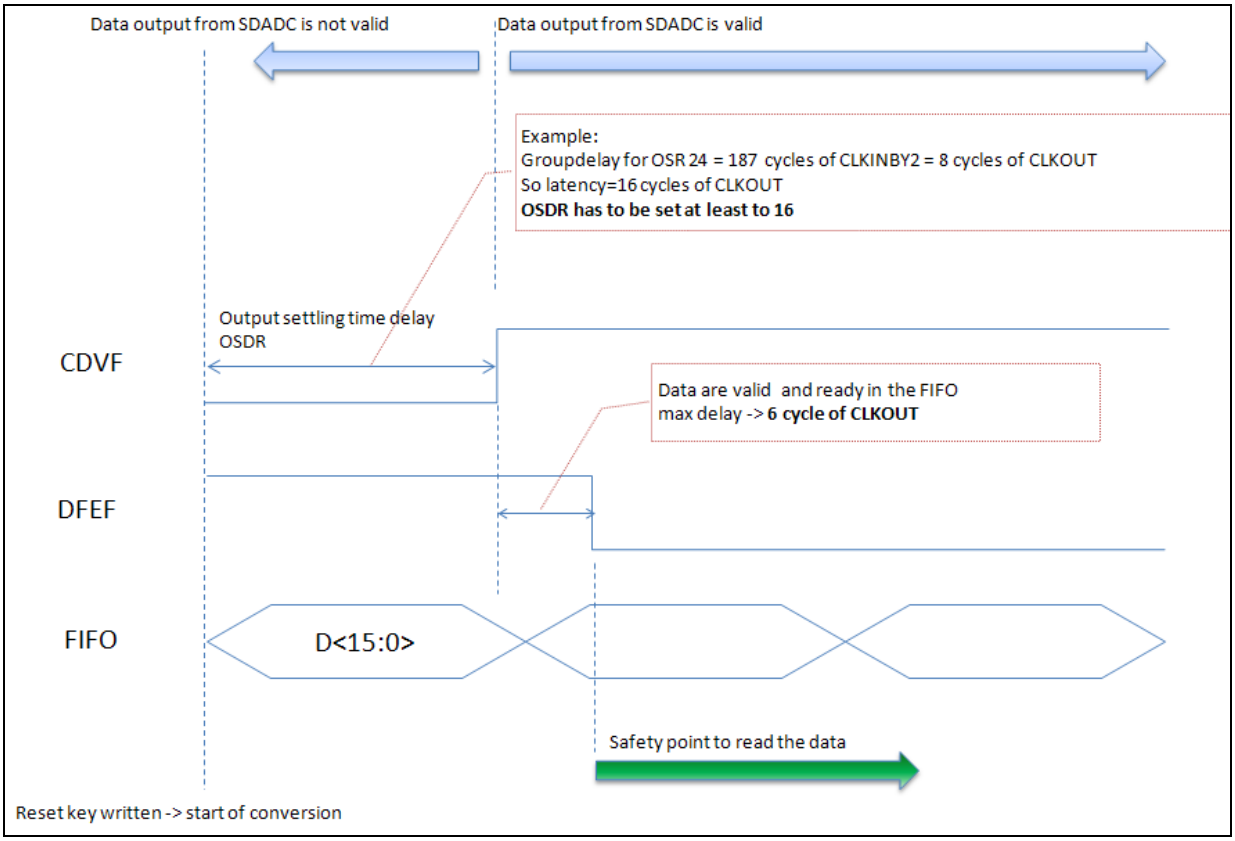
The time elapsed between a reset event and a Read data by the CDATA field of the CDR register are:

- Reset event ÷ Data Valid flag(CDVS) = output settling time delay (OSDR has to be set at least to 16)
- Data Valid flag ÷ Data FIFO is not empty = 0.5*CLK_Out + 3*CLK_In
- Data FIFO is not empty ÷ read Data (safety point) = 2*CLK_In

where:

CLK_Out = CLK_In/(2*OSR)

Figure 9. SDADC data conversion time



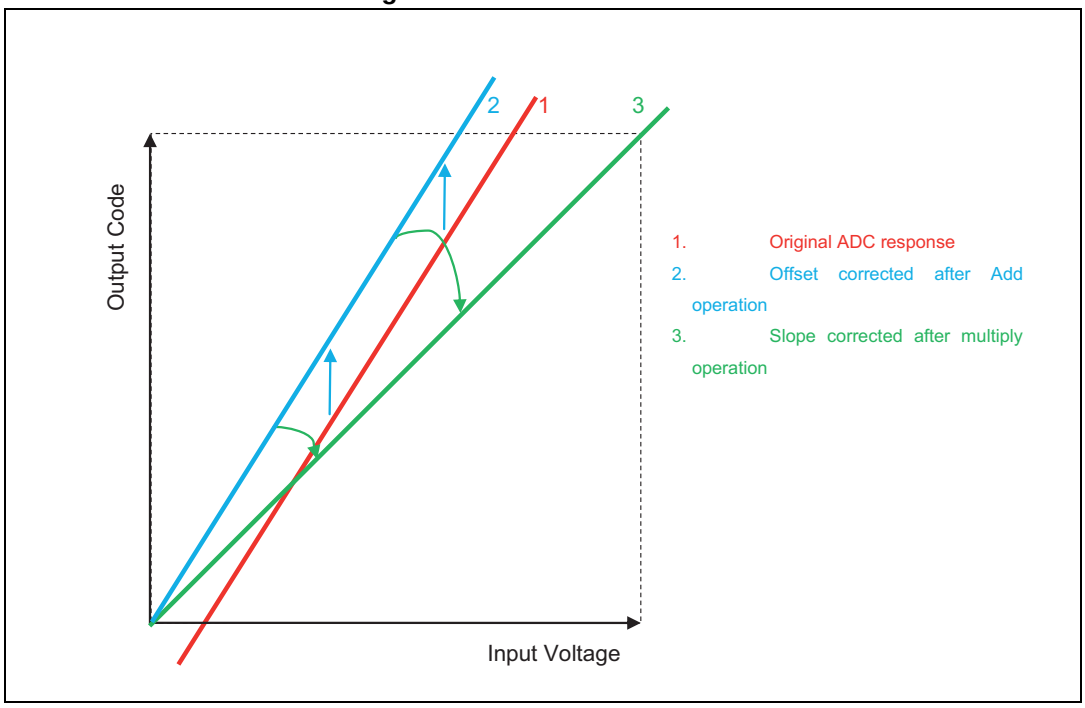
2.7 SDADC calibration

A gain error can be evaluated after an offset calibration. To have a calibrated output value:

- Get calibration OffSet: $OffSet_{Calibration}$
- Get calibration Gain: $Gain_{Calibration}$
- Apply the retrieved value to the output ADC value: ADC_{Value}

$$ADC_{Calibrated} = \frac{(ADC_{Value} + OffSet_{Calibration})}{Gain_{Calibration}}$$

Figure 10. SDADC calibration



2.8 Offset calibration procedure

To perform offset calibration, the following sequence must be applied:

1. Select differential mode of operation by writing MCR[MODE] to ‘0’
2. Configure the mux selection ANCHSEL field of CSR to ‘100’ or ‘101’ as required
   a) CSR = ‘100’ in case of data conversion after calibration in “single ended mode with negative input = VSS_HV_ADR_D”
   b) CSR = ‘101’ in case of data conversion after calibration in “differential mode” and “single ended mode with negative input = (VDD_HV_ADR_D - VSS_HV_ADR_D)/2”
3. Disable the bias on all the input analog channels by writing ENBIAS field of CSR to 0x00.
4. Disable the high pass filter by deasserting MCR[HPFEN]
5. Generate a reset event by writing 0x5AF0 to RESET_KEY of RKR
6. Read the digital output stored in FIFO after the output settling time
7. The measured offset can be used to nullify the offset error in the digital output. Expected output is 0b00_0000_0000_0000. The SDADC offset can be calculated as:

$$\text{Offset} = \text{Expected Output} - \text{Actual Output}$$

The offset must be calculated for each PGAN field setting since it is expected to vary with the gain configuration of SDADC.

2.9 Gain calibration step

To perform a gain calibration, the following sequence needs to be applied:

1. Select the differential mode of operation by writing MCR[MODE] to ‘0’.
2. Enable the Accurate Gain Error Mode Enable by writing MCR[GECEN] to ‘1’.
3. Configure the mux selection ANCHSEL field of CSR to ‘110’. This configuration will apply VREFP on positive terminal and VREFN on negative terminal.
4. Disable the bias on all the input analog channels by writing ENBIAS field of CSR to 0x00.
5. Disable the high-pass filter by deasserting MCR[HPFEN].
6. Generate a reset event by writing 0x5AF0 to the RESET_KEY field of RKR register.
7. Read the digital output (Dp) stored in FIFO after the output settling time. Expected output if there is no gain error is 0b0111_1001_1001_1001, corresponding to full positive scale after attenuation inserted by the internal filter (1*0.95). The measurement should be repeated to reduce contribution of noise during calibration process. Dp is the average value of attenuated positive full scale given by Dp = AVERAGE(CDR[CDATA])
8. Change the mux selection ANCHSEL field of CSR to ‘111’. This configuration will apply VREFN on positive terminal and VREFP on negative terminal.
9. Generate a reset event by writing 0x5AF0 to RESET_KEY of RKR.
10. Read the digital output (Dn) stored in the FIFO after the output settling time. Expected output if there is no gain error is 0b1000_0110_0110_0110, corresponding to the full negative scale after attenuation inserted by the internal filter (-1*0.95). The measurement should be repeated to reduce contribution of noise during the calibration process. Dn is the average value of an attenuated negative full scale given by Dn = AVERAGE(CDR[CDATA])
11. The SDADC error can be calculated as:
    Gain = (Dp – Dn) / 2^{16}
12. The measured gain value can be used to nullify the gain errors in the digital output.

During the calibration, the number of full scale conversion (Dp, Dn) is directly correlated to the rejection of noise.

The value conversion is depending on the application noise. It is recommended to run at least 16 conversions before calculating the average value.

For a calibrated conversion, the data CDR[CDATA] provided by the SDADC should be normalized using the calculated gain:

$$CDATA_{norm} = CDR[CDATA]/Gain$$

3 SAR vs SD

In *Table 2* a short comparison between the two kinds of ADC.

Table 2. SAR ADC vs SD ADC

SAR ADC	SD ADC
+ Fast signal acquisition	+ High dynamic signals
+ Fast input signal multiplexing	+ Strongly monotonic by design
+ DC signals	+ Fast AC signals
- Resolution is limited	+ High resolution possible with less effort
- Demanding in terms of technology (matching of components parameters)	- Input signal switching takes time

4 ADC: automotive use cases

In *Table 3* a short comparing between the two kinds of ADC, automotive use cases.

Table 3. ADC automotive use cases

SAR ADC	SD ADC
Signal acquisition – Temperature: water, oil, air – Pressure: fuel, oil, air – Position: flaps for air inlet and outlet, actuators, sensors – Human interfaces: switches, pedals, levers – Movement: acceleration, deceleration	**Combustion parameters** – Knock detection – Exhaust gas analysis
Diagnosis – Sensors and actors – Supply monitoring	**Battery management** – Cell balancing inside smart lithium ion batteries
	Audio & Video signals – Entertainment – Voice to command

Appendix A

It is supposed that the peripherals had got a reset default configuration. For a full specification refer to the SPC574Kxx reference manual and datasheet (see *Section B.1: Reference documents*).

Whenever a register or a register bit field is referred, the following notation applies:

```
PERIPHERAL_NAME.REG_NAME[index].R           //register access
PERIPHERAL_NAME.REG_NAME[index].B.BIT_NAMEx //bit field
```

where,

- *index* is used if more than one register with the same functionality exist within a peripheral (for example a timer with 16 channels, then it would have 16 control registers for each channel one);
- *x* is used when more than one bit with the same functionality exists within a register (e.g. 5 chip select enable bits). Peripheral name may be omitted when the context is obvious.

For example: `SDACD_0.MCR.B.EN`, refers to the EN bit from MCR register in ADADC_0 peripheral.

A.1 Offset calibration procedure

Figure 11. Offset calibration procedure

```
float SDADC_Offset_Cal(volatile struct SDADC_tag *SDADC, int SDADC_dev){
    vuint16_t x = 0;
    int16_t ADC_Data_N = 0;
    int16_t Offset = 0;
    int16_t Offset_Vect[17] = 0;
    float Offset_Average = 0;

    SDADC->SFR.B.DFORF = 1;          // RESET Data FIFO Overrun Flag
    SDADC->SFR.B.DFFF = 1;           // RESET Data FIFO Full Flag
    SDADC->FCR.B.FE = 1;             //FIFO data is one data.*/
    SDADC->MCR.B.EN = 1;             // Power up SDADC
    SDADC->MCR.B.FRZ = 0;            // Stop the SARADC conversions at the end of current
    channel conversion

    SDADC->OSDR.R = 0x000000FF;      // Set Output Settling Delay
    SDADC->MCR.B.PGAN = 0;           // Programmable Gain: Gain = 1
    SDADC->MCR.B.GECEN = 1;          // Disable gain error calibration mode
    SDADC->MCR.B.MODE = 0;           // Select differential mode
    SDADC->CSR.B.ANCHSEL = 4;        // in case of data conversion after calibration in
    "differential mode" and "single
                                     // ended mode with negative input = 0.5*VDD_HV_ADR_D"

    SDADC->CSR.B.BIASEN = 0x00;      // Disable the bias on all input analog channels
    SDADC->MCR.B.HPFEN = 0x0;        // Disable the high pass filter

    SDADC->RKR.B.RESET_KEY = 0x5AF0;// Generate a reset event
    while(SDADC->SFR.B.DFEF == 1);  // while FIFO is empty

    while((x<17) && (SDADC->SFR.B.DFEF == 0)){  // FIFO is not empty
        ADC_Data_N = SDADC->CDR.B.CDATA;    // the ADC_Data  will be ugual to 0b
        00_0000_0000_0000.
        Offset = (0 - ADC_Data_N);  // Offset = Expected Output - Actual Output
        Offset_Vect[x] = Offset ;   // save Offset in the vector
        printf("Offset[%d] -> SDACD_%d = %d \n",x,SDADC_dev,Offset);
        x++;
    };

    for (x=1;x<17;x++)
        Offset_Average += Offset_Vect[x];

    Offset_Average = Offset_Average/16;
    printf("Offset_AVERAGE -> SDACD_%d = %f \n",SDADC_dev,Offset_Average);

    SDADC->MCR.B.EN = 0;         // Power down SDADC
    return (Offset_Average);
}
```

4.1 Gain calibration procedure

Figure 12. Gain calibration procedure

```
float SDADC_Gain_Cal(volatile struct SDADC_tag *SDADC, int SDADC_dev){
    vint16_t ADC_Data_P = 0, ADC_Data_N = 0, x = 0;
    int16_t Data_P_Vect[17] = 0, Data_N_Vect[17] = 0;
    double Data_P_Average = 0, Data_N_Average = 0;
    vint32_t Gain = 0;
    float Gain_f = 0;

    SDADC->SFR.B.DFORF = 1;          // RESET Data FIFO Overrun Flag
    SDADC->SFR.B.DFFF = 1;           // RESET Data FIFO Full Flag
    SDADC->FCR.B.FE = 1;             // FIFO data is one data.*/
    SDADC->MCR.B.EN = 1;             // Power up SDADC
    SDADC->MCR.B.MODE = 0;           // Select differential mode
    SDADC->MCR.B.GECEN = 1;          // Enable gain error calibration mode
    SDADC->CSR.B.ANCHSEL = 6;        // Configure the mux selection: This configuration will
                          // apply VREFP on positive terminal and VREFN on negative terminal
    SDADC->CSR.B.BIASEN = 0x00;      // Disable the bias on all input analog channels
    SDADC->MCR.B.HPFEN = 0x0;        // Disable the high pass filter
    SDADC->RKR.B.RESET_KEY = 0x5AF0;     // Generate a reset event
    while(SDADC->SFR.B.DFEF == 1);   // while FIFO is empty
    SDADC->SFR.B.DFFF = 1;           // RESET Data FIFO Full Flag
    while((x<16) && (SDADC->SFR.B.DFEF == 0)){   // FIFO is not empty
        Data_P_Vect[x] = SDADC->CDR.B.CDATA;     // Converted Data Register
        printf("Data_P_Vect[%d] -> SDACD_%d = %d \n",x,SDADC_dev,Data_P_Vect[x]);
        x++;
    };
    for (x=1;x<16;x++)
        Data_P_Average += Data_P_Vect[x];
    ADC_Data_P = (vint16_t)(Data_P_Average/15);
    // the ADC_Data  will be ugual to 0b 0111_1111_1111_1111.
    printf("Data_P_Average -> SDACD_%d = %d \n",SDADC_dev,ADC_Data_P);
    SDADC->CSR.B.ANCHSEL = 7;        // Configure the mux selection: This configuration will
                          // apply VREFN on positive terminal and VREFP on negative terminal
    SDADC->RKR.B.RESET_KEY = 0x5AF0;// Generate a reset event
    while(SDADC->SFR.B.DFEF == 1);   // while FIFO is empty
    SDADC->SFR.B.DFFF = 1;           // RESET Data FIFO Full Flag
    x = 0;
    while((x<17) && (SDADC->SFR.B.DFEF == 0)){   // FIFO is not empty
        Data_N_Vect[x] = SDADC->CDR.B.CDATA;     // Converted Data Register
        printf("Data_N_Vect[%d] -> SDACD_%d = %d \n",x,SDADC_dev,Data_N_Vect[x]);
        x++;
    };
    for (x=1;x<17;x++)
        Data_N_Average += Data_N_Vect[x];
    ADC_Data_N = (vint16_t)(Data_N_Average/16);
    // the ADC_Data  will be ugual to 0b 1000_0000_0000_0000.
    printf("Data_N_Average -> SDACD_%d = %d \n",SDADC_dev,ADC_Data_N);

    Gain = ADC_Data_P - ADC_Data_N;
    Gain_f = (float) (((float) Gain)/65536);     // Calculated Gain
    printf("Gain SDACD_%d = %d/65536 = %f \n",SDADC_dev,Gain , Gain_f );
    SDADC->MCR.B.EN = 0;             // Power down SDADC
    return (Gain_f);
}
```

4.2 SDADC_0 CH2 setting

Figure 13. SDADC_0 CH2 setting

```
void SDADC0_CH2_Set(void){
    SDADC_0.SFR.B.DFORF = 1;          // RESET Data FIFO Overrun Flag
    SDADC_0.SFR.B.DFFF = 1;           // RESET Data FIFO Full Flag
    SDADC_0.FCR.B.FE = 0;             // Disable FIFO data */

    SDADC_0.MCR.B.EN = 1;             // Power up SDADC
    SDADC_0.OSDR.R = 23;              // Configure OSDR 0
    SDADC_0.MCR.B.MODE = 1;           // Single-ended input mode selected
    SDADC_0.MCR.B.VCOMSEL = 0;        // 0= Negative input terminal is biased with VREFN
                                      // 1= Negative input terminal is biased with VREFP/2
    SDADC_0.MCR.B.GECEN = 1;          // Enable gain error calibration mode
    SDADC_0.MCR.B.HPFEN = 0x0;        // Disable the high pass filter
    SDADC_0.MCR.B.FRZ = 0;            // Stop the SDADC conversions at the end of the current
    channel
    SDADC_0.MCR.B.PGAN = 0x0;         // Set gain of channel to 1
    SDADC_0.CSR.B.ANCHSEL = 2;        // Select analog channel 2 of SD-0
                                      // SD0_CH2 is on PB[0] <=> Motherboard Potentiometer RV1
                                      // This configuration will apply VREFN on negative terminal
    SDADC_0.CSR.B.BIASEN = 0x00;      // Disable the bias on all input analog channels

    SDADC_0.RKR.B.RESET_KEY = 0x5AF0;    // Generate a reset event
    while(SDADC_0.SFR.B.DFEF == 1);      // while FIFO is empty
    SDADC_0.SFR.B.DFFF = 1;              // RESET Data FIFO Full Flag
}
```

4.3 Test N.1: SDADC0_ CH2 acquisition

The follow example shows the SDADC0_CH0 acquisition with and without calibration.

The code in MAIN program is:

Figure 14. Test N.1: SDADC0_ CH2 Acquisition

```
/* SDADC_0_CH2 --- AN2 --- on Port PB[0] */
SIUL2.MSCR[16].B.WPDE = 1;
SIUL2.MSCR[16].B.IBE = 1;
SIUL2.MSCR[16].B.APC = 1;

SDADC_Offset_Cal_CH0 = SDADC_Offset_Cal(&SDADC_0, 0);
SDADC_Gain_Cal_CH0  = SDADC_Gain_Cal(&SDADC_0, 0);

printf("\n \n \n");
printf("************************************************* \n");
printf("************    K2 SDADC Calibration    *********** \n");
printf("***                                               *** \n");
printf("***   SDADC_0_CH2  -  AN2  -  on Port PB[0]       *** \n");
printf("***                                               *** \n");
printf("************************************************* \n");
printf("\n \n \n");

printf("Offset Calibration = %f \n",SDADC_Offset_Cal_CH0);
printf("Cain Calibration = %f \n\n",SDADC_Gain_Cal_CH0);

SDADC0_CH2_Set();   // Set SDADC_0_CH2

for (i=0; i<25; i++){
    SDADC_0_CH2 = (vint16_t)SDADC_0.CDR.B.CDATA;
    SDADC_0.SFR.B.DFFF = 1;          // RESET Data FIFO Full Flag
    CH2_OffSet = (float)(SDADC_0_CH2 + SDADC_Offset_Cal_CH0);
    CH2_Gain = (float)(CH2_OffSet / SDADC_Gain_Cal_CH0);
    CH2_Vcal = (float)((CH2_Gain/32768)*5);
    CH2_Vncal = (float)(((float)SDADC_0_CH2/32768)*5);
    printf("CH2= %d; CH2+Offset= %f (CH2+Offset)/G= %f Vcal=  %f  Vncal=  %f\n",SDADC_0_CH2
    CH2_OffSet,CH2_Gain, CH2_Vcal, CH2_Vncal);
}
```

4.3.1 CH2 acquisition values: 5V

By the board potentiometer, the input voltage on CH2 has been set to 5 V.

With this setting, the results are:

Figure 15. CH2 acquisition values: 5V

```
************************************************
************    K2 SDADC Calibration    ************
***                                              ***
***   SDADC_0_CH2  -  AN2  -  on Port PB[0]      ***
***                                              ***
************************************************


Offset Calibration = -54.312500
Cain Calibration = 0.946579

CH2= 31033; CH2+Offset= 30978.687500 (CH2+Offset)/G= 32726.998047 Vcal=  4.993743  Vncal=  4.735260
CH2= 31029; CH2+Offset= 30974.687500 (CH2+Offset)/G= 32722.771484 Vcal=  4.993099  Vncal=  4.734650
CH2= 31036; CH2+Offset= 30981.687500 (CH2+Offset)/G= 32730.166016 Vcal=  4.994227  Vncal=  4.735718
CH2= 31047; CH2+Offset= 30992.687500 (CH2+Offset)/G= 32741.787109 Vcal=  4.996000  Vncal=  4.737396
CH2= 31048; CH2+Offset= 30993.687500 (CH2+Offset)/G= 32742.843750 Vcal=  4.996161  Vncal=  4.737549
CH2= 31049; CH2+Offset= 30994.687500 (CH2+Offset)/G= 32743.900391 Vcal=  4.996323  Vncal=  4.737701
CH2= 31035; CH2+Offset= 30980.687500 (CH2+Offset)/G= 32729.109375 Vcal=  4.994066  Vncal=  4.735565
CH2= 31027; CH2+Offset= 30972.687500 (CH2+Offset)/G= 32720.658203 Vcal=  4.992776  Vncal=  4.734344
CH2= 31041; CH2+Offset= 30986.687500 (CH2+Offset)/G= 32735.449219 Vcal=  4.995033  Vncal=  4.736481
CH2= 31038; CH2+Offset= 30983.687500 (CH2+Offset)/G= 32732.279297 Vcal=  4.994549  Vncal=  4.736023
CH2= 31032; CH2+Offset= 30977.687500 (CH2+Offset)/G= 32725.941406 Vcal=  4.993582  Vncal=  4.735107
CH2= 31044; CH2+Offset= 30989.687500 (CH2+Offset)/G= 32738.617188 Vcal=  4.995517  Vncal=  4.736938
CH2= 31046; CH2+Offset= 30991.687500 (CH2+Offset)/G= 32740.730469 Vcal=  4.995839  Vncal=  4.737244
CH2= 31034; CH2+Offset= 30979.687500 (CH2+Offset)/G= 32728.052734 Vcal=  4.993905  Vncal=  4.735413
CH2= 31048; CH2+Offset= 30993.687500 (CH2+Offset)/G= 32742.843750 Vcal=  4.996161  Vncal=  4.737549
CH2= 31039; CH2+Offset= 30984.687500 (CH2+Offset)/G= 32733.335938 Vcal=  4.994711  Vncal=  4.736176
CH2= 31034; CH2+Offset= 30979.687500 (CH2+Offset)/G= 32728.052734 Vcal=  4.993905  Vncal=  4.735413
CH2= 31044; CH2+Offset= 30989.687500 (CH2+Offset)/G= 32738.617188 Vcal=  4.995517  Vncal=  4.736938
CH2= 31049; CH2+Offset= 30994.687500 (CH2+Offset)/G= 32743.900391 Vcal=  4.996323  Vncal=  4.737701
CH2= 31028; CH2+Offset= 30973.687500 (CH2+Offset)/G= 32721.714844 Vcal=  4.992938  Vncal=  4.734497
CH2= 31040; CH2+Offset= 30985.687500 (CH2+Offset)/G= 32734.392578 Vcal=  4.994872  Vncal=  4.736328
CH2= 31032; CH2+Offset= 30977.687500 (CH2+Offset)/G= 32725.941406 Vcal=  4.993582  Vncal=  4.735107
CH2= 31034; CH2+Offset= 30979.687500 (CH2+Offset)/G= 32728.052734 Vcal=  4.993905  Vncal=  4.735413
CH2= 31024; CH2+Offset= 30969.687500 (CH2+Offset)/G= 32717.488281 Vcal=  4.992292  Vncal=  4.733887
CH2= 31049; CH2+Offset= 30994.687500 (CH2+Offset)/G= 32743.900391 Vcal=  4.996323  Vncal=  4.737701
```

4.3.2 CH2 acquisition values: 0 - 5V

By the board potentiometer, the input voltage on CH2 has been set from 0 V to 5 V.

With this setting, the results are:

Figure 16. CH2 acquisition values: 0 - 5V

```
************************************************
************    K2 SDADC Calibration    ************
***                                              ***
***   SDADC_0_CH2  -  AN2  -  on Port PB[0]      ***
***                                              ***
************************************************


Offset Calibration = -54.562500
Cain Calibration = 0.946518

CH2= 54; CH2+Offset= -0.562500 (CH2+Offset)/G= -0.594284 Vcal=  -0.000091  Vncal=  0.008240
CH2= 51; CH2+Offset= -3.562500 (CH2+Offset)/G= -3.763796 Vcal=  -0.000574  Vncal=  0.007782
CH2= 60; CH2+Offset= 5.437500 (CH2+Offset)/G= 5.744740 Vcal=  0.000877  Vncal=  0.009155
CH2= 60; CH2+Offset= 5.437500 (CH2+Offset)/G= 5.744740 Vcal=  0.000877  Vncal=  0.009155
CH2= 38; CH2+Offset= -16.562500 (CH2+Offset)/G= -17.498348 Vcal=  -0.002670  Vncal=  0.005798
CH2= 60; CH2+Offset= 5.437500 (CH2+Offset)/G= 5.744740 Vcal=  0.000877  Vncal=  0.009155
CH2= 45; CH2+Offset= -9.562500 (CH2+Offset)/G= -10.102819 Vcal=  -0.001542  Vncal=  0.006866
CH2= 775; CH2+Offset= 720.437500 (CH2+Offset)/G= 761.145081 Vcal=  0.116142  Vncal=  0.118256
CH2= 3137; CH2+Offset= 3082.437500 (CH2+Offset)/G= 3256.607666 Vcal=  0.496919  Vncal=  0.478668
CH2= 6633; CH2+Offset= 6578.437500 (CH2+Offset)/G= 6950.145508 Vcal=  1.060508  Vncal=  1.012115
CH2= 8340; CH2+Offset= 8285.437500 (CH2+Offset)/G= 8753.597656 Vcal=  1.335693  Vncal=  1.272583
CH2= 9422; CH2+Offset= 9367.437500 (CH2+Offset)/G= 9896.735352 Vcal=  1.510122  Vncal=  1.437683
CH2= 11151; CH2+Offset= 11096.437500 (CH2+Offset)/G= 11723.430664 Vcal=  1.788854  Vncal=  1.701508
CH2= 12450; CH2+Offset= 12395.437500 (CH2+Offset)/G= 13095.829102 Vcal=  1.998265  Vncal=  1.899719
CH2= 14256; CH2+Offset= 14201.437500 (CH2+Offset)/G= 15003.875977 Vcal=  2.289410  Vncal=  2.175293
CH2= 16095; CH2+Offset= 16040.437500 (CH2+Offset)/G= 16946.787109 Vcal=  2.585875  Vncal=  2.455902
CH2= 18250; CH2+Offset= 18195.437500 (CH2+Offset)/G= 19223.552734 Vcal=  2.933281  Vncal=  2.784729
CH2= 20423; CH2+Offset= 20368.437500 (CH2+Offset)/G= 21519.335938 Vcal=  3.283590  Vncal=  3.116302
CH2= 22644; CH2+Offset= 22589.437500 (CH2+Offset)/G= 23865.832031 Vcal=  3.641637  Vncal=  3.455200
CH2= 24801; CH2+Offset= 24746.437500 (CH2+Offset)/G= 26144.710938 Vcal=  3.989366  Vncal=  3.784332
CH2= 27390; CH2+Offset= 27335.437500 (CH2+Offset)/G= 28880.000000 Vcal=  4.406738  Vncal=  4.179382
CH2= 30499; CH2+Offset= 30444.437500 (CH2+Offset)/G= 32164.669922 Vcal=  4.907939  Vncal=  4.653778
CH2= 31043; CH2+Offset= 30988.437500 (CH2+Offset)/G= 32739.408203 Vcal=  4.995637  Vncal=  4.736786
CH2= 31030; CH2+Offset= 30975.437500 (CH2+Offset)/G= 32725.673828 Vcal=  4.993542  Vncal=  4.734802
CH2= 31047; CH2+Offset= 30992.437500 (CH2+Offset)/G= 32743.634766 Vcal=  4.996282  Vncal=  4.737396
```

4.4 Test N.2: SDADC0_ CH2 acquisition with bias

The following example shows the SDADC0_CH0 acquisition with bias on negative input. This means that the:

negative input = (VDD_HV_ADR_D - VSS_HV_ADR_D)/2.

4.4.1 SDADC_BIAS_Set

To set the bias on negative terminal, to enable the VCOMSEL, and respective BIASEN field:

Figure 17. SDADC_BIAS_Set

```
void SDADC_BIAS_set(volatile struct SDADC_tag *SDADC, int SDADC_CH){
    SDADC_0.SFR.B.DFORF = 1;         // RESET Data FIFO Overrun Flag
    SDADC_0.SFR.B.DFFF = 1;          // RESET Data FIFO Full Flag
    SDADC_0.MCR.B.EN = 1;            // Power up SDADC
    SDADC_0.MCR.B.VCOMSEL = 1;       // 0= Negative input terminal is biased with VREFN
                                     // 1= Negative input terminal is biased with VREFP/2
    SDADC_0.CSR.B.ANCHSEL = SDADC_CH;        // Select analog SDADC_CH of SDADC0
    SDADC_0.CSR.B.BIASEN = (1<<SDADC_CH);    // Enable the bias on SDADC_CH input analog channels

    SDADC_0.RKR.B.RESET_KEY = 0x5AF0;    // Generate a reset event
    while(SDADC_0.SFR.B.DFEF == 1);      // while FIFO is empty
    SDADC_0.SFR.B.DFFF = 1;          // RESET Data FIFO Full Flag
}
```

4.4.2 CH2 acquisition values: -2.5 V - +2.5 V

The acquisition results, show the -2.5 V for an input pin value at 0 V, and +2.5 V for input pin value at 5 V.

The acquired values are not by regular steps, because the increase of the input value is obtained via a potentiometer rotated by hand.

Figure 18. CH2 acquisition values: -2.5 V - +2.5 V

```
************************************************
************    K2 SDADC Calibration    ************
***                                              ***
***   SDADC_0_CH2  -  AN2  -  on Port PB[0]      ***
***                                              ***
************************************************


Offset Calibration = -56.687500
Cain Calibration = 0.946487

CH2= 56; CH2+Offset= -0.687500 (CH2+Offset)/G= -0.726370 Vcal=  -0.000111  Vncal=  0.008545
CH2= -15447; CH2+Offset= -15503.687500 (CH2+Offset)/G= -16380.236328 Vcal=  -2.499426  Vncal=  -2.357025
CH2= -15458; CH2+Offset= -15514.687500 (CH2+Offset)/G= -16391.857422 Vcal=  -2.501199  Vncal=  -2.358704
CH2= -15452; CH2+Offset= -15508.687500 (CH2+Offset)/G= -16385.519531 Vcal=  -2.500232  Vncal=  -2.357788
CH2= -15492; CH2+Offset= -15548.687500 (CH2+Offset)/G= -16427.781250 Vcal=  -2.506680  Vncal=  -2.363892
CH2= -13880; CH2+Offset= -13936.687500 (CH2+Offset)/G= -14724.640625 Vcal=  -2.246802  Vncal=  -2.117920
CH2= -10658; CH2+Offset= -10714.687500 (CH2+Offset)/G= -11320.475586 Vcal=  -1.727368  Vncal=  -1.626282
CH2= -7665; CH2+Offset= -7721.687500 (CH2+Offset)/G= -8158.256836 Vcal=  -1.244851  Vncal=  -1.169586
CH2= -4968; CH2+Offset= -5024.687500 (CH2+Offset)/G= -5308.773438 Vcal=  -0.810055  Vncal=  -0.758057
CH2= -3104; CH2+Offset= -3160.687500 (CH2+Offset)/G= -3339.386719 Vcal=  -0.509550  Vncal=  -0.473633
CH2= -1744; CH2+Offset= -1800.687500 (CH2+Offset)/G= -1902.494873 Vcal=  -0.290298  Vncal=  -0.266113
CH2= -222; CH2+Offset= -278.687500 (CH2+Offset)/G= -294.443939 Vcal=  -0.044929  Vncal=  -0.033875
CH2= 1056; CH2+Offset= 999.312500 (CH2+Offset)/G= 1055.811646 Vcal=  0.161104  Vncal=  0.161133
CH2= 2571; CH2+Offset= 2514.312500 (CH2+Offset)/G= 2656.466797 Vcal=  0.405345  Vncal=  0.392303
CH2= 4094; CH2+Offset= 4037.312500 (CH2+Offset)/G= 4265.574219 Vcal=  0.650875  Vncal=  0.624695
CH2= 5756; CH2+Offset= 5699.312500 (CH2+Offset)/G= 6021.540527 Vcal=  0.918814  Vncal=  0.878296
CH2= 7558; CH2+Offset= 7501.312500 (CH2+Offset)/G= 7925.422363 Vcal=  1.209324  Vncal=  1.153259
CH2= 9433; CH2+Offset= 9376.312500 (CH2+Offset)/G= 9906.430664 Vcal=  1.511601  Vncal=  1.439362
CH2= 11661; CH2+Offset= 11604.312500 (CH2+Offset)/G= 12260.397461 Vcal=  1.870788  Vncal=  1.779327
CH2= 14055; CH2+Offset= 13998.312500 (CH2+Offset)/G= 14789.750000 Vcal=  2.256737  Vncal=  2.144623
CH2= 15489; CH2+Offset= 15432.312500 (CH2+Offset)/G= 16304.825195 Vcal=  2.487919  Vncal=  2.363434
CH2= 15509; CH2+Offset= 15452.312500 (CH2+Offset)/G= 16325.956055 Vcal=  2.491143  Vncal=  2.366486
CH2= 15488; CH2+Offset= 15431.312500 (CH2+Offset)/G= 16303.769531 Vcal=  2.487758  Vncal=  2.363281
CH2= 15505; CH2+Offset= 15448.312500 (CH2+Offset)/G= 16321.730469 Vcal=  2.490499  Vncal=  2.365875
CH2= 15490; CH2+Offset= 15433.312500 (CH2+Offset)/G= 16305.881836 Vcal=  2.488080  Vncal=  2.363586
```

Appendix B Other information

B.1 Reference documents

- *SPC574Kxx - 32-bit Power Architecture® based MCU for automotive applications* (RM0334, Doc ID 023671)
- *32-bit Power Architecture® based MCU for automotive applications* (SPC574K72E5, SPC574K72E7 Datasheet, Doc ID 023601)

B.2 Acronyms

Table 4. Acronyms

Acronym	Name
SDADC	Sigma-Delta Analog-to-Digital Converter
SARADC	Successive Approximation Analog-to-Digital Converter

Revision history

Table 5. Document revision history

Date	Revision	Changes
02-Dec-2013	1	Initial release.

Please Read Carefully:

Information in this document is provided solely in connection with ST products. STMicroelectronics NV and its subsidiaries (“ST”) reserve the right to make changes, corrections, modifications or improvements, to this document, and the products and services described herein at any time, without notice.

All ST products are sold pursuant to ST’s terms and conditions of sale.

Purchasers are solely responsible for the choice, selection and use of the ST products and services described herein, and ST assumes no liability whatsoever relating to the choice, selection or use of the ST products and services described herein.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted under this document. If any part of this document refers to any third party products or services it shall not be deemed a license grant by ST for the use of such third party products or services, or any intellectual property contained therein or considered as a warranty covering the use in any manner whatsoever of such third party products or services or any intellectual property contained therein.

UNLESS OTHERWISE SET FORTH IN ST’S TERMS AND CONDITIONS OF SALE ST DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY WITH RESPECT TO THE USE AND/OR SALE OF ST PRODUCTS INCLUDING WITHOUT LIMITATION IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE (AND THEIR EQUIVALENTS UNDER THE LAWS OF ANY JURISDICTION), OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT.

ST PRODUCTS ARE NOT DESIGNED OR AUTHORIZED FOR USE IN: (A) SAFETY CRITICAL APPLICATIONS SUCH AS LIFE SUPPORTING, ACTIVE IMPLANTED DEVICES OR SYSTEMS WITH PRODUCT FUNCTIONAL SAFETY REQUIREMENTS; (B) AERONAUTIC APPLICATIONS; (C) AUTOMOTIVE APPLICATIONS OR ENVIRONMENTS, AND/OR (D) AEROSPACE APPLICATIONS OR ENVIRONMENTS. WHERE ST PRODUCTS ARE NOT DESIGNED FOR SUCH USE, THE PURCHASER SHALL USE PRODUCTS AT PURCHASER’S SOLE RISK, EVEN IF ST HAS BEEN INFORMED IN WRITING OF SUCH USAGE, UNLESS A PRODUCT IS EXPRESSLY DESIGNATED BY ST AS BEING INTENDED FOR “AUTOMOTIVE, AUTOMOTIVE SAFETY OR MEDICAL” INDUSTRY DOMAINS ACCORDING TO ST PRODUCT DESIGN SPECIFICATIONS. PRODUCTS FORMALLY ESCC, QML OR JAN QUALIFIED ARE DEEMED SUITABLE FOR USE IN AEROSPACE BY THE CORRESPONDING GOVERNMENTAL AGENCY.

Resale of ST products with provisions different from the statements and/or technical features set forth in this document shall immediately void any warranty granted by ST for the ST product or service described herein and shall not create or extend in any manner whatsoever, any liability of ST.

ST and the ST logo are trademarks or registered trademarks of ST in various countries.
Information in this document supersedes and replaces all information previously supplied.
The ST logo is a registered trademark of STMicroelectronics. All other names are the property of their respective owners.

© 2013 STMicroelectronics - All rights reserved

STMicroelectronics group of companies

Australia - Belgium - Brazil - Canada - China - Czech Republic - Finland - France - Germany - Hong Kong - India - Israel - Italy - Japan - Malaysia - Malta - Morocco - Philippines - Singapore - Spain - Sweden - Switzerland - United Kingdom - United States of America

www.st.com