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

1  SAR ADC characteristics ............................................ 6
   1.1 Features .................................................... 7
   1.2 Considerations on the input impedance of the signal source .... 8
   1.3 Conversion timings .......................................... 10
   1.4 SARADC: constrain and timing ............................... 11

2  SD ADC characteristics ............................................. 13
   2.1 Features .................................................... 14
   2.2 Considerations on the input impedance of the signal source .... 14
   2.3 SDADC and DMA ........................................... 15
   2.4 SDADC conversion timing .................................... 15
   2.5 Data conversion step ......................................... 17
   2.6 Data conversion timing .................................... 17
   2.7 SDADC calibration ........................................... 18
   2.8 Offset calibration procedure ................................. 19
   2.9 Gain calibration step ....................................... 20

3  SAR vs SD .......................................................... 21

4  ADC: automotive use cases ......................................... 22

Appendix A ............................................................. 23
   A.1 Offset calibration procedure ................................. 24
   A.2 Gain calibration procedure ................................. 25
   A.2 SDADC_0 CH2 setting ....................................... 26
   4.3 Test N.1: SDADC0_ CH2 acquisition ...................... 26
      4.3.1 CH2 acquisition values: 5V ............................ 27
      4.3.2 CH2 acquisition values: 0 - 5V ..................... 28
   4.4 Test N.2: SDADC0_ CH2 acquisition with bias .......... 29
      4.4.1 SDADC_BIAS_Set ..................................... 29
      4.4.2 CH2 acquisition values: -2.5 V - +2.5 V .......... 30

Appendix B  Other information......................................... 31
B.1  Reference documents ............................................. 31
B.2  Acronyms .......................................................... 31

Revision history ......................................................... 32
List of tables

Table 1. 10 bit SARADC timing ................................................................. 11
Table 2. SAR ADC vs SD ADC................................................................. 21
Table 3. ADC automotive use cases ....................................................... 22
Table 4. Acronyms .................................................................................. 31
Table 5. Document revision history ....................................................... 32
List of figures

Figure 1. SAR ADC block architecture ................................................................. 6
Figure 2. SARADC block diagram ................................................................. 7
Figure 3. SAR ADC input equivalent circuit ................................................................. 8
Figure 4. Input equivalent circuit (Fast SARn channels) ...................................................... 9
Figure 5. Input equivalent circuit (SARB channels) .............................................................. 9
Figure 6. SDADC block architecture ................................................................. 13
Figure 7. SDADC input equivalent circuit ................................................................. 14
Figure 8. Conversion delay ......................................................................................... 16
Figure 9. SDADC data conversion time ........................................................................ 18
Figure 10. SDADC calibration ...................................................................................... 19
Figure 11. Offset calibration procedure ........................................................................ 24
Figure 12. Gain calibration procedure ............................................................................ 25
Figure 13. SDADC_0 CH2 setting ............................................................................... 26
Figure 14. Test N.1: SDADC0_CH2 Acquisition .......................................................... 27
Figure 15. CH2 acquisition values: 5V .......................................................................... 28
Figure 16. CH2 acquisition values: 0 - 5V ................................................................. 29
Figure 17. SDADC_BIAS_Set ..................................................................................... 29
Figure 18. CH2 acquisition values: -2.5 V - +2.5 V ...................................................... 30
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
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
The sampling capacitor $C_s$ 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 $C_s$ equal to 6 pF a resistance of 417 kΩ ($R_{eq} = 1/(f_c C_s)$) is obtained.

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

---

**Figure 4. Input equivalent circuit (Fast SARn channels)**

- $R_{SW1}$: Channel Selection Switch Impedance
- $R_{AD}$: Sampling Switch Impedance
- $C_P$: Pin Capacitance (two contributions, $C_{P1}$ and $C_{P2}$)
- $C_S$: Sampling Capacitance
- $R_{CMSW}$: Common mode switch
- $R_{CML}$: Common mode resistive ladder

**Figure 5. Input equivalent circuit (SARB channels)**

- $R_{SW1}$: Channel Selection Switch Impedance (two contributions $R_{SW1}$ and $R_{SW2}$)
- $R_{AD}$: Sampling Switch Impedance
- $C_P$: Pin Capacitance (three contributions, $C_{P1}$, $C_{P2}$ and $C_{P3}$)
- $C_S$: Sampling Capacitance
- $R_{CMSW}$: Common mode switch
- $R_{CML}$: Common mode resistive ladder
of the SAR. In order to run
the SAR at the highest sampling speed, \( R_L \) should respect the following constraint:

\[
R_L < 350\text{Ohm}
\]

Of course, \( R_L \) shall be sized also according to the current limitation constraints, in
combination with \( R_S \) (source impedance) and \( R_F \) (filter resistance), due to some
consideration on Nyquist theorem, and transfer charge, then a constraints on \( C_S \) 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 SAMSEL bitfield of the
 corresponding channel data register.

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

The precharging phase duration is given by

\[
t_{\text{prechg}} = \text{PRECHG} \cdot t_{\text{ck}}
\]

where \( \text{PRECHG} \geq 2 \) (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:
tsample = INPSAMP * tck

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 * tck

t_{eval} = 21 * tck (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 tck refers to reciprocal of fck, where fck = 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

<table>
<thead>
<tr>
<th>Clock (MHz)</th>
<th>TCK (µs)</th>
<th>PRECHG</th>
<th>INPSAMP 10 bit (µs)</th>
<th>tPRECHG (µs)</th>
<th>tSAMPLE 10 bit (µs)</th>
<th>tEVAL 10 bit (µs)</th>
<th>tCONV 10 bit (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.500</td>
<td>2</td>
<td>4</td>
<td>1.000</td>
<td>2.000</td>
<td>10.500</td>
<td>13.500</td>
</tr>
<tr>
<td>4</td>
<td>0.250</td>
<td>2</td>
<td>4</td>
<td>0.500</td>
<td>1.000</td>
<td>5.250</td>
<td>6.750</td>
</tr>
<tr>
<td>6</td>
<td>0.167</td>
<td>2</td>
<td>4</td>
<td>0.333</td>
<td>0.667</td>
<td>3.500</td>
<td>4.500</td>
</tr>
<tr>
<td>8</td>
<td>0.125</td>
<td>3</td>
<td>5</td>
<td>0.375</td>
<td>0.625</td>
<td>2.625</td>
<td>3.625</td>
</tr>
<tr>
<td>10</td>
<td>0.100</td>
<td>3</td>
<td>6</td>
<td>0.300</td>
<td>0.600</td>
<td>2.100</td>
<td>3.000</td>
</tr>
<tr>
<td>12</td>
<td>0.083</td>
<td>4</td>
<td>7</td>
<td>0.333</td>
<td>0.583</td>
<td>1.750</td>
<td>2.667</td>
</tr>
</tbody>
</table>
### Table 1. 10 bit SARADC timing (continued)

<table>
<thead>
<tr>
<th>Clock (MHz)</th>
<th>TCK (µs)</th>
<th>PRECHG</th>
<th>INPSAMP 10 bit (µs)</th>
<th>tPRECHG (µs)</th>
<th>tSAMPLE 10 bit (µs)</th>
<th>tEVAL 10 bit (µs)</th>
<th>tCONV 10 bit (µs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td>0.071</td>
<td>4</td>
<td>8</td>
<td>0.286</td>
<td>0.571</td>
<td>1.500</td>
<td>2.357</td>
</tr>
<tr>
<td>14.6</td>
<td>0.068</td>
<td>4</td>
<td>9</td>
<td>0.274</td>
<td>0.616</td>
<td>1.438</td>
<td>2.329</td>
</tr>
</tbody>
</table>
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:

- $R_{est} < 1 \, \text{k}\Omega$
- $C_{f} = 1\, \text{nF}$ to $10\, \text{nF}$
- $R_{f} = \text{max} \, 1 \, \text{k}\Omega$

With this value the cutoff frequency is:

$$f_{\text{cutoff}} = \frac{1}{2\pi \cdot R_{f} \cdot C_{f}}$$

for

- $C_{f} = 10 \, \text{nF}$
- $R_{f} = 1 \, \text{kOhm}$
- $f_{\text{cutoff}} \approx 15 \, \text{kHz}$

### 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

$$f_{d} = \frac{f_{S}}{2 \times \text{OSR}}$$

- $\delta_{\text{G}}$ is the Groupdelay. The Groupdelay depend on OSR value
- $t_{\text{STARTUP}}$ is the start-up time from the power down state. The ADC is “ready” EN is pulled to ‘1’. After $t_{\text{STARTUP}}$ 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 $\div$ Data Valid flag (CDVS) = output settling time delay (OSDR has to be set at least to 16)
- Data Valid flag $\div$ Data FIFO is not empty = $0.5^*\text{CLK}_\text{Out} + 3^*\text{CLK}_\text{In}$
- Data FIFO is not empty $\div$ read Data (safety point) = $2^*\text{CLK}_\text{In}$

where:
$$\text{CLK}_\text{Out} = \frac{\text{CLK}_\text{In}}{2^*\text{OSR}}$$
2.7 SDADC calibration

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

- Get calibration OffSet: OffSet\textsubscript{Calibration}
- Get calibration Gain: Gain\textsubscript{Calibration}
- Apply the retrieved value to the output ADC value: ADC\textsubscript{Value}

\[
\text{ADC}_{\text{Calibrated}} = \frac{\text{ADC}_{\text{Value}} + \text{OffSet}_{\text{Calibration}}}{\text{Gain}_{\text{Calibration}}}
\]
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:
    \[ \text{Gain} = \frac{Dp - Dn}{2^{16}} \]
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:

\[ \text{CDATAnorm} = \frac{\text{CDR[CDATA]}}{\text{Gain}} \]
3 SAR vs SD

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

<table>
<thead>
<tr>
<th>SAR ADC</th>
<th>SD ADC</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Fast signal acquisition</td>
<td>+ High dynamic signals</td>
</tr>
<tr>
<td>+ Fast input signal multiplexing</td>
<td>+ Strongly monotonic by design</td>
</tr>
<tr>
<td>+ DC signals</td>
<td>+ Fast AC signals</td>
</tr>
<tr>
<td>- Resolution is limited</td>
<td>+ High resolution possible with less effort</td>
</tr>
<tr>
<td>- Demanding in terms of technology (matching of components parameters)</td>
<td>- Input signal switching takes time</td>
</tr>
</tbody>
</table>
4 **ADC: automotive use cases**

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

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

Table 3. ADC automotive use cases
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_NAME x //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

```c
float SDADC_Offset_Cal(void) {
    SDADC_B.DF = 81; // RESET Data FIFO
    SDADC_B.DFF = 81; // RESET Data FIFO Full Flag
    SDADC_B.PE = 1; // FIFO data is one data.*
    SDADC_B.EN = 1; // Power up SDADC
    SDADC_B.FRZ = 81; // Stop the SARADC conversions at the end of current
    channel conversion
    SDADC_B.R = 0X00000000; // Set Output Settling Delay
    SDADC_B.PGA = 0; // Programmable Gain: Gain = 1
    SDADC_B.GEC = 1; // Disable gain error calibration mode
    SDADC_B.MODR = 1; // Select differential mode
    SDADC_B.ANSR = 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_B.BIAS = 0; // Disable the bias on all input analog channels
    SDADC_B.HPF = 0; // Disable the high pass filter
    SDADC_B.RST = 0x5AF8; // Generate a reset event
    while(SDACD->SFR.B.DF = 1); // while FIFO is empty
    while((x<17) && (SDADC->SFR_B.DF = 0)){ // FIFO is not empty
        ADC_DATA_N = SDADC->SFR_B.CDATA; // the ADC_Data will be equal to 0b
        00_0000_0000_0000
        Offset = (ADC_Data_N); // Offset = Expected Output - Actual Output
        Offset_Vect[x] = Offset; // save Offset in the vector
        printf("Offset[%d] -> SDACD%d = %.2f \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 = %.2f \n", SDADC_dev, Offset_Average);
    SDADC_B.EN = 0; // Power down SDADC
    return (Offset_Average);
}
```
4.1 Gain calibration procedure

Figure 12. Gain calibration procedure

```c
float SDADC_Gain_Cal(volatile struct SDADC_tag *SDADC, int SDADC_dev){
    uint16_t ADC_Data_P = 0, ADC_Data_N = 0, x = 0;
    int16_t Data_P_Vect[16] = 0, Data_N_Vect[16] = 0;
    double Data_P_Average = 0, Data_N_Average = 0;
    uint32_t Gain = 0;
    float Gain_f = 0;

    SDADC->SPR.B.DFORW = 1; // RESET Data FIFO Overrun Flag
    SDADC->SPR.B.DFFF = 1; // RESET Data FIFO Full Flag
    SDADC->FCR.B.RF = 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.GECRN = 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.BIASRN = 0x00; // Disable the bias on all input analog channels
    SDADC->MCR.B.HFPPN = 0x00; // Disable the high pass filter
    SDADC->RRK.B.RESET_KEY = 0x59F0; // Generate a reset event
    while((SDADC->SPR.B.DFEEF == 1)); // while FIFO is empty
    SDADC->SPR.B.DFFF = 1; // RESET Data FIFO Full Flag
    while((x<16) & (SDADC->SPR.B.DFEEF == 0)){ // FIFO is not empty
        Data_P_Average += Data_P_Vect[x];
        // the ADC Data will be usual to 0b 0_1111_1111_1111_1111.
        printf("Data_P_Average -> SDADC_%d = %.f \n",x,SDADC_dev,Data_P_Vect[x]);
    x++;
    }
    for (x=1;x<6;x++)
    {
        Data_P_Average += Data_P_Vect[x];
        ADC_Data_P = (uint16_t)(Data_P_Average/16);
        // the ADC Data will be usual to 0b 0000_0000_0000_0000_0000.
        printf("Data_P_Average -> SDADC_%d = %.f \n",x,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->RRK.B.RESET_KEY = 0x59F0; // Generate a reset event
    while((SDADC->SPR.B.DFEEF == 1)); // while FIFO is empty
    SDADC->SPR.B.DFFF = 1; // RESET Data FIFO Full Flag
    x = 0;
    while((x<17) & (SDADC->SPR.B.DFEEF == 0)){ // FIFO is not empty
        Data_N_Vect[x] = SDADC->CDR.B.CDATA; // Converted Data Register
        printf("Data_N_Vect[%d] -> SDADC_%d = %.f \n",x,SDADC_dev,Data_N_Vect[x]);
    x++;
    }
    for (x=1;x<7;x++)
    {
        Data_N_Average += Data_N_Vect[x];
        ADC_Data_N = (uint16_t)(Data_N_Average/16);
        // the ADC_Data will be usual to 0b 1000_0000_0000_0000.
        printf("Data_N_Average -> SDADC_%d = %.f \n",x,SDADC_dev,ADC_Data_N);
    }
    Gain = ADC_Data_P - ADC_Data_N;
    Gain_f = (float)((float)Gain)/65536; // Calculated Gain
    printf("Gain SDADC_%d = %.f \n",SDADC_dev,Gain , Gain_f );
    SDADC->MCR.B.EN = 0; // Power down SDADC
    return (Gain_f);
}
```
4.2 SDADC_0 CH2 setting

The following example shows the SDADC0 CH0 acquisition with and without calibration.

void SDADC0_CH2_Set(void){
    SDADC_0.SFRB.DFORP = 1;  // RBST Data FIFO Overrun Flag
    SDADC_0.SFRB.DFFF = 1;  // RBST Data FIFO Full Flag
    SDADC_0.PCRB.FE = 0;  // Disable FIFO data */
    SDADC_0.MCRB.EN = 1;  // Power up SDADC
    SDADC_0.OSDRB.R = 23;  // Configure OSDR 0
    SDADC_0.MCRB.MODE = 1;  // Single-ended input mode selected
    SDADC_0.MCRB.VCOMSEL = 0;  // 0- Negative input terminal is biased with VREF
                               // 1- Negative input terminal is biased with VREFP/2
    SDADC_0.MCRB.GECN = 1;  // Enable gain error calibration mode
    SDADC_0.MCRB.HDFEN = 0x0;  // Disable the high pass filter
    SDADC_0.MCRB.FRZ = 0;  // Stop the SDADC conversions at the end of the current channel
    SDADC_0.MCRB.PGAIN = 0x0;  // Set gain of channel to 1
    SDADC_0.CSRB.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.CSRB.BIASLEN = 0x00;  // Disable the bias on all input analog channels
    SDADC_0.RKR.B.RESET_KEY = 0x5A50;  // Generate a reset event
    while(SDADC_0.SFRB.DFEF == 1);  // While FIFO is empty
    SDADC_0.SFRB.DFFF = 1;  // RESET Data FIFO Full Flag
}

4.3 Test N.1: SDADC0_ CH2 acquisition

The following example shows the SDADC0_0 acquisition with and without calibration.

The code in MAIN program is:
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:
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:
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:

```
void SDADC_BIAS_set(volatile struct SDADC_tag *SDADC, int SDADC_CH) {
    SDADC_0.SFR.B.DPORP = 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 VREFP
    SDADC_0.CSR.B.ANCHESEL = SDADC_CH; // Select analog SDADC_CH of SDADC0
    SDADC_0.CSR.B.BIASEN = (<<SDADC_CH); // Enable the bias on SDADC_CH input analog channels
    SDADC_0.RKR.B.RESET_KEY = 0x5A0F; // Generate a reset event
    while(SDADC_0.SFR.B.DFFF == 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
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>
<thead>
<tr>
<th>Acronym</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDADC</td>
<td>Sigma-Delta Analog-to-Digital Converter</td>
</tr>
<tr>
<td>SARADC</td>
<td>Successive Approximation Analog-to-Digital Converter</td>
</tr>
</tbody>
</table>
# Revision history

Table 5. Document revision history

<table>
<thead>
<tr>
<th>Date</th>
<th>Revision</th>
<th>Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>02-Dec-2013</td>
<td>1</td>
<td>Initial release.</td>
</tr>
</tbody>
</table>

