

## ADC BEHAVIOR OF STM 300 / STM 3xy – HOW TO MEASURE RAIL TO RAIL Rail-to-Rail Sensor Applications using the STM 300 and STM 33x

### 1. Introduction

STM 300 is a universal, customer configurable EnOcean device platform for various sensor applications. While STM 33x is calibrated and configured as temperature sensor / set point control and STM 32x as a magnetic sensor contact, STM 300 can be used more flexible for different other sensors, e.g. light sensor. Similar to the other above mentioned ready to use sensors, STM 300 devices can be customer specific programmed/configured. In order to enable Rail-to-Rail measurements, special firmware called STMSEN (source code see DolphinAPI examples, [1.]) and appropriate external HW adaption circuit as shown in the following must be used\*.

\* The Dolphin chip AD converter inside STM 3xy cannot measure real Rail-to-Rail (e.g. full scale, GND to SWPWR). For details, please consult Dolphin Core Description and specific STM 3xy user manuals [1.], [1.], **Error! Reference source not found.**

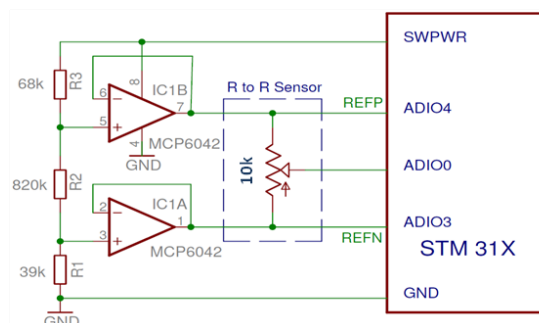
#### 1.1. References

- [1.] Dolphin Core Description - (<http://www.enocean.com/knowledge-base/>)
- [2.] EnOcean Development Tools Download - (<http://www.enocean.com/en/download/>)
- [3.] EnOcean Support - (<http://www.enocean.com/support/>, [support@enocean.com](mailto:support@enocean.com))
- [4.] STM 300 Product page - ([http://www.enocean.com/en/enocean\\_modules/stm-300/](http://www.enocean.com/en/enocean_modules/stm-300/))
- [5.] STM 330 Product page - ([http://www.enocean.com/en/enocean\\_modules/stm-330/](http://www.enocean.com/en/enocean_modules/stm-330/))

To enable Rail-to-Rail measurements, two "offset references", REFN and REFP – which are within the measurement range of the Dolphin chip – must be provided first by creating a simple appropriate voltage divider. For more accuracy, these can be buffered by voltage followers. This external circuit "shifts" the supply voltage rails of the device into the measurement range of the internal ADC converter. The provided offset voltages are defined by the divider R1, R2 and R3 accordingly specified Dolphin chip analog input range limitations. The intended sensor adapter circuit (in its simplest form simulated just like a 10 k pot below) must be designed to work and deliver at full scale voltages within the specified REFN (0, about 4.2% of SWPWR) and REFP (about 92.67% of SWPWR) ADC rails. Two additional analog inputs (in this case ADIO3 and ADIO4) are used to read first these external references. All results are then reported to those two references.

### 2. Rail-to-Rail external sensor application on STM 3xy platforms.

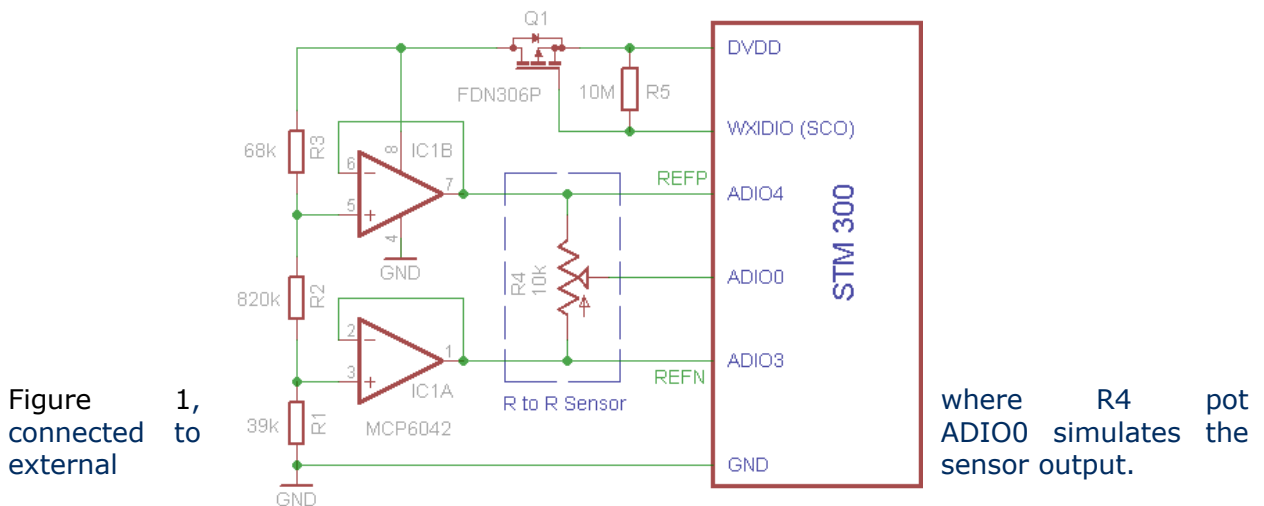
**Note:** Analogue inputs ADIO3/ADIO4 are not used on default STM 3xy FW, means the default firmware in STM 3xy does not perform any measurements on ADIO3/ADIO4. Therefore, the original STM 3xy device must be reprogrammed (over its 20 pin connector) with the above mentioned STMSEN FW before implementing REFP and REFN external circuit.



ADC BEHAVIOR OF STM 300 / STM 3xy – HOW TO MEASURE RAIL TO RAIL  
 Rail-to-Rail Sensor Applications using the STM 300 and STM 33x

**3. Rail-to-Rail external sensor applications on STM 300 platforms.**

By using a STM 300 HW platform (with STMSSEN FW) in an application requiring Rail-to-Rail, supply sensor voltage is typically provided by DVDD. This is switched on only during the measurement driven by the SCO (WXIDIO) control pin and Q1. See reference design in



**Figure 1 Reference design for STM 300, Rail-to-Rail measurement on ADIO0**

The default STM 300 FW does not support Rail-to-Rail Measurements [4.]. As internal reference for measurements, the STM 300 default FW uses:

- RVDD - RF supply voltage regulator output (1.8 V)
- RVSS - Ground connection for RF (0 V)

To enable “quite” Rail-to-Rail measurements, we have to measure as first the new REFN and REFP references on ADIO pins, like STMSSEN FW does.

STM 300 default FW has no free ADIO pins to measure external references during analog measurements. You have to find an appropriate I/O pair and free it accordingly in the firmware. As analog input, only following ADIO pins are actually provided:

- ADIO\_0

## ADC BEHAVIOR OF STM 300 / STM 3xy – HOW TO MEASURE RAIL TO RAIL Rail-to-Rail Sensor Applications using the STM 300 and STM 33x

- ADIO\_1
- ADIO\_2
- ADIO\_3
- ADIO\_4

ADIO\_0, ADIO\_1 and ADIO\_2 are already used in STM 300 default FW as analog input channels (0, 1 and 2) while ADIO\_3 and ADIO\_4 are used as digital input channels (0, 1). To enable Rail-to-Rail, you have to use the ADIO\_3 and ADIO\_4 for REFN and REFP (recommended). This way you cannot use them any more as digital input channels 0 and 1.

By using a configuration pin as digital input, you have to set the configuration values in DolphinStudio and write them into the configuration area, overwriting the hardware settings. Otherwise, if the configuration area does not hold any values, the module will try to read the hardware configuration pins, what could result into non-defined states. Optionally, you can change the STM 300 default FW to take another default value when the configuration area does not hold any set values.

### 3.1 Step by step, Rail-to-Rail Sensor Application with STM 300

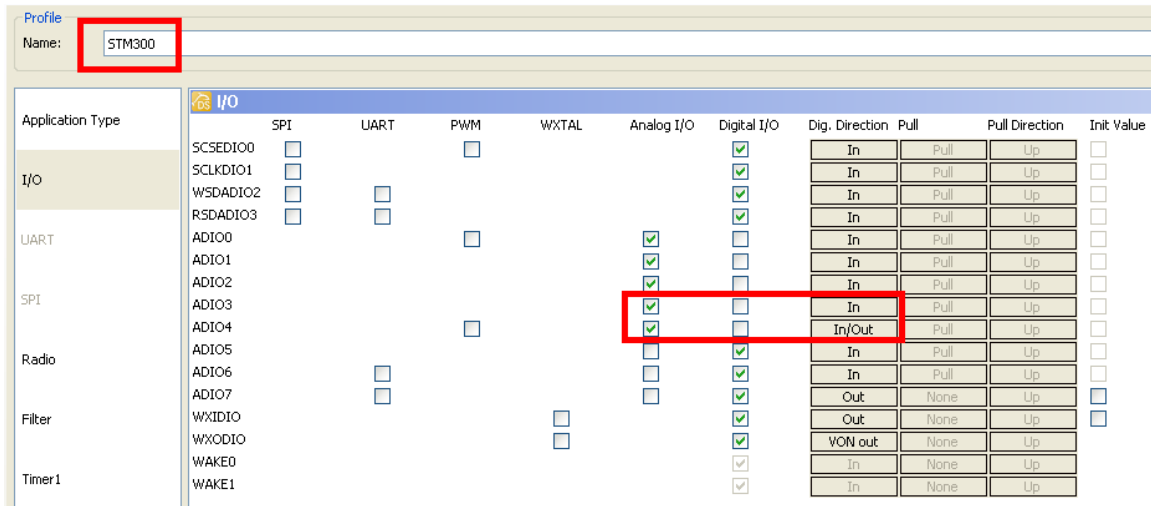
To enable Rail-to-Rail measurements on STM 300 platforms, you need to perform following steps:

1. Use the reference design as presented above.
2. Change Source Code of STM 300 FW to use ADIO\_3 and ADIO\_4 as references during analog measurements. Change the function - void MeasureAnalog() - located in file measurement.c

C:\Projekte\CVS_STM300_DB\SourceCode\measurement.c	* C:\Projekte\CVS_STM300_DB\SourceCode\measurement_rar.c
<pre> <b>void MeasureAnalog() Original STM 300 File</b> {     sint16 s16adio0value;     sint16 s16adio1value;     sint16 s16adio2value;     uint16 u16Value;      // do the measurement of the 3 analog ports     io_enableAnalog(1); // configuring hardware     for analog measurement no VGA, increase current     to Analog current     //lint -save -e534     io_ulpMeasAnalog(GPIO1_ADIO_0,     &amp;s16adio0value); // measure ADIO_0 against     internal voltage     io_ulpMeasAnalog(GPIO1_ADIO_1,     &amp;s16adio1value); // measure ADIO_1 against     internal voltage     if (unigcfg.cfg.stm300_cfg.u8AD_2_Mode ==     FLASH_ERASED_8)         io_ulpMeasAnalog(GPIO1_ADIO_2,     &amp;s16adio2value); // measure ADIO_2 against     internal voltage     else         io_ulpMeasAnalog(VDD_4, &amp;s16adio2value);     // measure VDD_4 against internal voltage and     put this in AD_2 value     io_ulpMeasAnalog(RVSS, &amp;s16gnegref); //     measure negative reference against internal     voltage     io_ulpMeasAnalog(RVDD, &amp;s16gposref); //     measure positive reference against internal     voltage     //lint -restore     io_enableAnalog(0); // restoring previous     configuration for radio functionality and reduce     power consumption </pre>	<pre> <b>void MeasureAnalog() Changed STM 300 File</b> {     sint16 s16adio0value;     sint16 s16adio1value;     sint16 s16adio2value;     uint16 u16Value;      // do the measurement of the 3 analog ports     io_enableAnalog(1); // configuring hardware     for analog measurement no VGA, increase current     to Analog current     //lint -save -e534     io_ulpMeasAnalog(GPIO1_ADIO_0,     &amp;s16adio0value); // measure ADIO_0 against     internal voltage     io_ulpMeasAnalog(GPIO1_ADIO_1,     &amp;s16adio1value); // measure ADIO_1 against     internal voltage     if (unigcfg.cfg.stm300_cfg.u8AD_2_Mode ==     FLASH_ERASED_8)         io_ulpMeasAnalog(GPIO1_ADIO_2,     &amp;s16adio2value); // measure ADIO_2 against     internal voltage     else         io_ulpMeasAnalog(VDD_4, &amp;s16adio2value);     // measure VDD_4 against internal voltage and     put this in AD_2 value     io_ulpMeasAnalog(ADIO_3, &amp;s16gnegref); //     measure negative reference against internal     voltage     io_ulpMeasAnalog(ADIO_4, &amp;s16gposref); //     measure positive reference against internal     voltage     //lint -restore     io_enableAnalog(0); // restoring previous     configuration for radio functionality and reduce     power consumption </pre>

ADC BEHAVIOR OF STM 300 / STM 3xy – HOW TO MEASURE RAIL TO RAIL  
 Rail-to-Rail Sensor Applications using the STM 300 and STM 33x

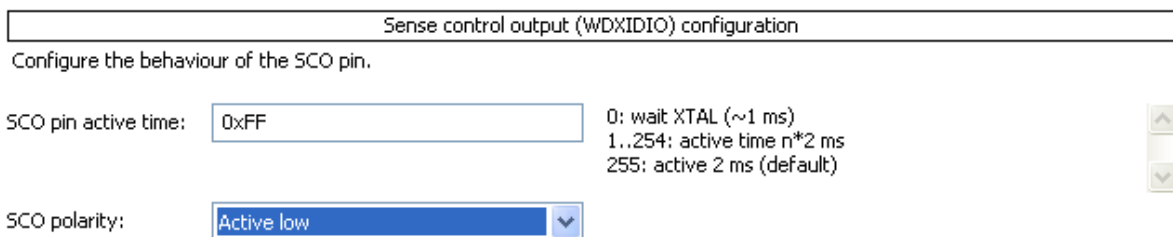
3. Change the STM 300 pin configuration in DolphinStudio of ADIO\_3 and ADIO\_4 from digital inputs to analog inputs, as described below.
  - a. Open the stm300.dat file in DolphinStudio. (e.g.



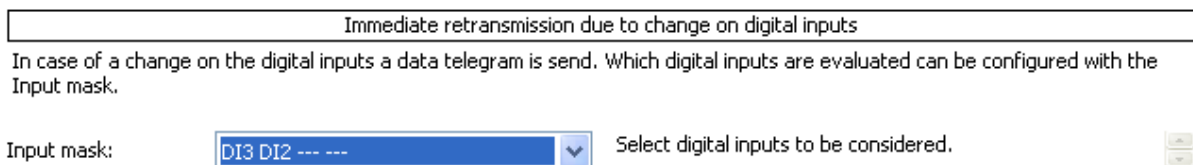
.\SourceCode\Profiles\STM300.dat )

Change the pin configuration from digital to analog input.

- b. Save the profile (stm300.dat) file and generate the configuration files.
4. Compile and flash the new FW into the STM 300 device.
5. Configure the STM 300 accordingly with DolphinStudio Tool. Set SCO pin as active LOW (if you are using active HIGH convert polarity of SCO in reference design)



6. Set the Digital INPUT mask so DI0 and DI1 are not considered. For example see figure below, also other combinations are available.



7. Write the configurations to the module.

## ADC BEHAVIOR OF STM 300 / STM 3xy – HOW TO MEASURE RAIL TO RAIL Rail-to-Rail Sensor Applications using the STM 300 and STM 33x

---

### **Disclaimer**

The information provided in this document is subject to modifications and intended for reference purposes only. EnOcean assumes no liability either for violation of industrial property or other rights of third parties that stem from this information, nor for errors and / or omissions.

We reserve the right to make changes without prior notice. Make sure that this is the latest version of the document before use. For the latest documentation, visit the EnOcean website at [www.enocean.com](http://www.enocean.com)