PICOSTRAIN®

Technical White Paper

Construction Guideline for Solar-driven Scales

07 October, 2008
Document-No. WP001 V2.0

Introduction

Did you ever see a solar scale based on metal strain gauges that does not require an additional battery? It is unlikely because this is not possible with current standard technologies. In times of climate change and renewable energy discussion, completely solar driven devices are of great interest and have above-average market opportunities.

Do strain gauge electronics exist that might close the gap? Yes they do! The PS08 is the latest component from acam-messelectronic gmbh and has a very low current consumption thanks to its measuring principle. It therefore allows the construction of solar scales with excellent measurement precision.

The PS08 is the latest development in the PICOSTRAIN series of devices which was established successfully in 2004. The strength of the PS08 is that it combines the PICOSTRAIN method with a microcontroller and an LCD driver. As a result, a true single-chip solution for measuring strain gauges is now available and opens new fields of applications.

This white paper shows the status quo in the field of solar scales, current and future solutions as well as issues of concern related to the design of such a scale. It contains a step-by-step guide to build a solar scale by means of the PS08.

Author: Ralf Emberger

Index

Current Situation

In the introduction, we posed a question about the availability of pure solar scales based on strain gauges. The few products which are available require a support battery at minimum or they use a completely different measuring principle such as capacitance measurement.

The classical approach for building a scale follows. The force is measured by means of strain gauges mounted on a load cell. The strain gauge resistors are wired in a Wheatstone bridge configuration. The differential output voltage is amplified and digitized by an analog-to-digital converter (ADC). The signal is further processed by a microprocessor (µP) and displayed on an LCD. The problem is that the sensor is permanently supplied with current (about 3 to 5 mA). Other components such as the ADC, µC and LCD need an additional 1 mA. Taking into account that a 20 cm² solar panel offers only about 50 to 60 µA under the condition of good office light, it becomes obvious that it is not possible to build a battery-less solar scale.

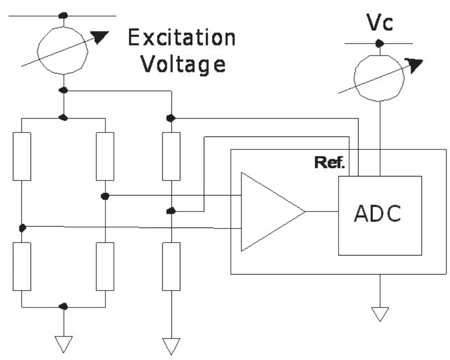


Fig. 1: Classical Wheatstone bridge with preamplifier and A/D-converter

Some design approaches are based on loading an accumulator while the light intensity is high and taking current from the accumulator when the lighting conditions are bad. As an alternative, batteries are connected in parallel to the solar panel. Only a small portion of the power comes from the solar panel. The great majority of current demand is supplied by the battery.

There is a chance to build pure solar scales if the sensor is not a strain gauge but a capacitive sensor. In this case the force varies the distance of two plates that form a capacitor, where a microprocessor measures the capacitance variation. Buffer the current with an electrolytic capacitor and it is then possible to operate solely from solar power.

Capacitive scales are similar in production cost to strain gauge scales but show less measurement quality. Thanks to the higher resolution and linearity, strain gauge sensors became the standard solution for weight scales and currently cover more than 95% of the market.

In consequence, it would be great to have a solution for solar scales that operates with the standard metal strain gauge approach. The PSØ8, developed by acam, fills this need perfectly. In the following section you can read about the basic principle of the PICOSTRAIN method. The subsequent sections show how to build a solar scale with the PSØ8.

How does the PICOSTRAIN method work?

acam is specialized in high-precision time interval measurement. Based on this method it is possible to measure strain gauge sensors with a few µA of current consumption. Thereby, the measurement quality is higher than with current solutions. The keyword is PICOSTRAIN and describes a method for low-current strain gauge measurement.

PICOSTRAIN uses a new approach which provides significant advantages when compared to A/D-converter solutions. The resistance ratio is calculated from a time measurement instead of a voltage measurement. This time interval is measured with very high resolution and much less power dissipation.

The sensing resistors together with a capacitor form a low-pass filter (RC network). The capacitor is first charged up to the supply voltage. Then it is disconnected and discharged via the sensing resistor. The discharge time down to the trigger level of a comparator is measured by means of a TDC (time-to-digital converter). Typical discharge times are in the range of 30 to 140 µs and the precision of the measurement is about 15 ps in a single measurement with the PSØ8.

The two sensing resistors are measured in time multiplex at the same capacitor and comparator. While calculating the ratio, the absolute value of the capacitance and the comparator's trigger level are eliminated. There are further disturbances coming from the input resistance of the drivers (Rdson) or the delay of the comparator that are eliminated by patented circuits and algorithms inside the PSØ8. The final result is virtually free of gain error and very stable over temperature.

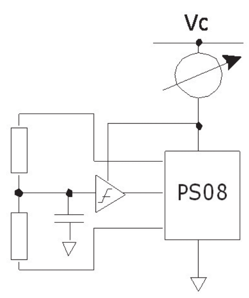


Fig. 2: Simplified diagram PSØ8

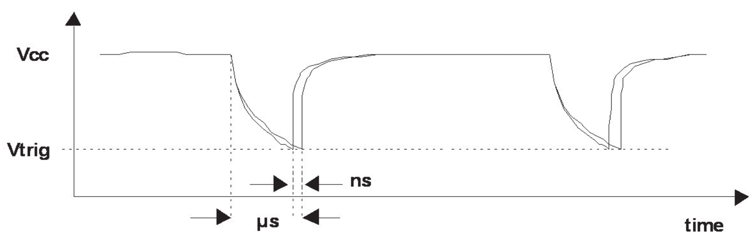


Fig. 3: The ratio of discharge times equals the ratio of the resistors

PICOSTRAIN does not need a full-bridge. Although full bridges can be measured, due to the measuring principle of PICOSTRAIN, a half bridge is sufficient. Therefore two half-bridges are used. The sensing resistors are connected directly to the converter. There is no need for a separate supply of the sensor. Thanks to the pulsed operation the current is easily controlled and far below the current of comparable A/D-converter solutions.

With the PS08 it is possible to reach the following benchmark data with respect to current consumption, maximum resolution and update rate: The current consumption may be reduced to 15 to 20 µA when properly set up for this goal (2,000 scale divisions at 1mV/V sensitivity and 3 Hz update rate). This low value is possible because of the pulsed manner of operation, the fact that the sensor is supplied directly from the chip and the reduced update rate. As a further advantage the self-heating of the sensor is minimized.

The maximum resolution of the PS08 is more than 19 bit effective for 2 mV/V sensitivity (common value with strain gauges) and 5 Hz update rate. Referring to the resistance, this means a resolution of more than 28 bit! The maximum update rate of the PS08 is 1,000 Hz. In general, the parameters, current consumption, resolution and update rate are interrelated and can be adjusted by configuration.

These benchmark data, especially the current consumption, clearly indicate that the realization of a purely solar powered scale is near at hand.

1.0 Preliminary Considerations

1.1 Application Specification

Before starting to build a solar scale the boundary conditions should be fixed. This includes fixed goals for the scale such as resolution and update rate but may include lighting conditions or the type of scale. The following list shows some important criteria:

- **Resolution**

What is the maximum load and with how many steps shall it be resolved? In this context we usually talk about "stable scale divisions". These are the steps with stable display.
Example:
10 kg are resolved in steps of 2 g which means 5,000 stable scale divisions. Internally it is necessary to have a resolution that is about 5 to 6 times better. In this case we will need 25,000 divisions. Alternatively: $2^{14.6}$ = 25,000, which leads to 14.6 bits of effective resolution.

- **Update rate**

The update rate relates how often a result is generated. An update rate of 3 Hz indicates that the result is updated 3 times per second. The internal sampling rate depends on the converter and the averaging rate and might be much higher (usually several kHz). Further, the higher the update rate the higher the current consumption and the lower the maximum possible resolution.

- **Solar panel**

The electricity yield depends on the size and type of solar cell. There are solar cells that are optimized for daylight, others specialize in artificial light. The bigger the solar panel the more current can be drawn out of it. A detailed discussion is given in the next section.

- **Lighting conditions**

Will the scale be exposed to bright lights typical of a workplace or reside in darkened environment typical of a bathroom? A postal scale may fall into the former category while a body scale may fall into the latter.

1.0 Preliminary Considerations

• Type of scale

The type of scale has direct influence on the above mentioned factors. Body scales have resolution and update rate data which are different from those of kitchen scales. Also the lighting conditions vary with the type of scale.

On the basis of these criteria it is possible to write a specification for the solar scale. The following example shows a set of features that will be transferred into a specification:

Body scale:

- 150 kg shall be resolved with 100 g
- Update of the measurement result minimum once per second, output on display
- Solar panel max. 4 cm x 5 cm for artificial light
- Lighting conditions: poor to medium lighting, total darkness as special case
- Scale shall switch on/off automatically

Specification:

- Resolution: 1500 stable scale divisions
- Update rate: > 1 Hz
- Solar panel size max. 20 cm²
- Current at artificial light, poor to medium, about 20 µA to 80 µA
- Cover situations with total darkness
- Auto-Off & Auto-On functionality

1.2 Selection of the solar cell

The selection of the solar cell is significant for the maximum available current. In general, bigger solar cell and better lighting conditions lead to more available current. On the other hand it is necessary to do worst-case investigations to see how much current is available under poor conditions. This information defines the minimum current level at which the system shall operate.

The relevant parameters for the solar panel selection are:

a) Size of the panel
b) Type of solar cell (outdoor / indoor)
c) Nominal operating current and nominal operating voltage
d) Expected lighting conditions

1.0 Preliminary Considerations

a) The light efficiency is typically better for large panels than small panels. However, the trend and transition is not linear. A comparison of solar panel from Sinonar [1] that are similar in construction shows the differences:

Type	Dimensions [mm]	Area [cm²]	Vop [V]	Iop [µA]	Current/Area [µA/cm²]
SS-6728-A	66.8 x 27.8	18,57	3	25	1,35
SS-5520	55.0 x 20.0	11	3	14	1,27
SS-5314-A	53.0 x 13.8	7,31	3	11	1,5
SS-4111	41.2 x 11.0	4,53	3	3,5	0,77

Tab. 1: Current versus Area for Sinonar [1] solar panels

b) There are varying types of solar cells which differ in construction and material.
But in general we have the following classification:

- Outdoor: Adopted to the solar light spectrum. Deliver several mA under direct sun light. These cells exhibit undesirable properties and very low current production when being operated in rooms with artificial light. They will probably not be used for solar scales.

- Indoor: Adopted to the artificial light spectrum. Under direct sunlight the efficiency is below that of outdoor cells, but is still in the range of mA. Therefore, this kind of solar cell would be ideal for building a solar scale. It offers sufficient current indoors and when used outdoors the lighting condition is typically much better than indoors. This allows it to produce sufficient current despite being removed from the ideal light spectrum.

c) Solar panels can be purchased for different voltages and output currents. There are solar cells [1] in the increments of 1.5 V, 1.8 V, 2.2 V, 3.0 V, up to 17.0 V. The related operating current varies from 2 µA up to 177 mA.

The performance data of outdoor solar panels is usually quoted to standard test conditions (STC). The conditions, light irradiation of 100 mW/cm², cell temperature of 25°C, radiation angle of 90°, and light spectrum of 1.5 AM (=Air Mass), are more or less nominal values and should be seen as laboratory conditions. For indoor cells the performance data usually refer to a specific illumination, given in lux. The performance data should be requested in any case from the manufacturer.

1.0 Preliminary Considerations

d) The lighting conditions have the most significant influence on the output current. On web pages [2] and [3] (noted in the appendix) there is a description which states that for **outdoor modules** the nominal power of a solar module has to be multiplied with a factor between 0.5 and 7, depending on the day conditions. This rule will not fit for **indoor solar** cells because they are adopted to artificial light and show a different behavior. Their behavior can be characterized by means of the physical measures of brightness or lighting condition.

The colloquial word 'brightness' is best expressed in a physical terms as 'illuminance' (luminous flux per distance). The following table shows some illumination data for different lighting conditions:

Bright sunny day:	100.000 lx
Shadow in summer:	10.000 lx
Clouded winter day:	3.500 lx
Office light:	750 lx
Corridor light:	100 lx

Tab. 2: Luminance [Lux] for several lighting conditions

A data sheet of Schott Company [4] shows the typical characteristics of an indoor solar panel versus illumination. Following this, an indoor solar panel with an area of 10 cm² gives the following output power:

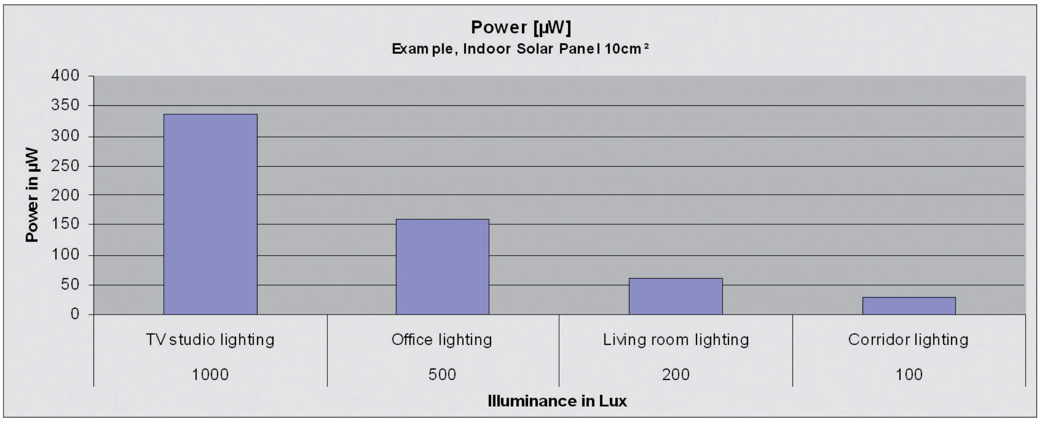


Fig. 4: Power in µW of an indoor solar cell versus illuminance

It is obvious from the diagram that the current yield varies by a large degree with differing lighting conditions. Therefore it is necessary to understand the relationship between the performance and the illumination of the indoor cell. The solar cells from Sinonar [1] are specified at 200 lx which are fairly bad lighting conditions. For sure it will give more safety in this theoretical discussion to assume lighting conditions that are worse than reality and to set up a system that fits to this minimum current.

In summary, from above parameter descriptions (a) to (d) it is obvious that several aspects have to be taken into account when selecting a solar panel. An important insight is that a solar panel with size of 10 to 20 cm² will provide only about 10 µA current under poor conditions. Classical electronic solutions with a quiescent current in the milliampere range definitely are no solution for building a solar scale.

1.3 Reducing Current Consumption

The high permanent current of a Wheatstone bridge in combination with an A/D converter is the most significant drawback for solar applications. According to Ohm's law the current into a Wheatstone bridge with four 1 kOhm resistors at 5 V is 5 mA or 3 mA at 3 V. Often the strain gauge resistors are only available in 350 Ohm and therefore the current is even higher.

PICOSTRAIN, in contrast, takes a different approach. The current reduction comes mainly from the different measurement principle and the sequence control that provides power to several portions of the circuit only when needed for the measurement.

Reduction by measuring principle: With PICOSTRAIN the strain gauge sensors are measured by means of discharge times. The sensors are driven in a pulsed manner rather than a continuous manner. Current flows into the sensors only during the discharge phase from the capacitor. The capacitor itself is recharged in the charging phase, as the following figure shows:

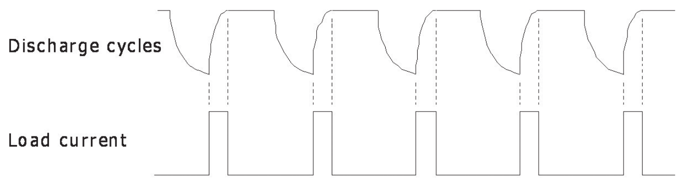


Fig. 5: Discharge cycles and load current

1.0 Preliminary Considerations

The necessary load current depends on the voltage swing, the capacitance and the number of charges. For the charge of a capacitor we have Q = C*U and with the typical values of a PICOSTRAIN measurement, 1.5 V swing and 100 nF capacitance, we get:

$Q = C \cdot U = 1{,}5\ V \cdot 100\ nF = 150\ nAs$ (Gl. 1)

The number of charge cycles defines the current into the strain gauge:

$I = 150\ nAs \cdot 100\ \text{Entladungen}/s = 15\ \mu A$ (Gl. 2)

The current into the sensor is very low compared to the current dissipated with the classical Wheatstone bridge approach. It may be further reduced by having fewer discharge cycles. The PS08 offers various operating modes that allow optimization. A continuous mode exists wherein the capacitor is steadily discharged. Also, a single conversion mode exists wherein the capacitor is discharged only a few times, followed by a period of no operation. During this period the system switches off units that are not in use. These include the the oscillator and the comparator. This gives a further reduction of current.

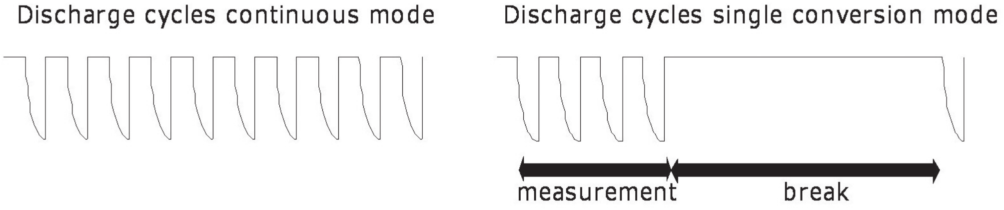


Fig. 6: Main operating modes

Reducing the number of discharge cycles will reduce the current consumption of the load capacitor. Therefore, halving the number of discharge cycles from 100 to 50 will halve the current into the load capacitor from 15 µA to 7.5 µA. Of course, reducing the number of discharge cycles will also reduce the resolution.

1.0 Preliminary Considerations

Further current drains include the oscillator, the digital logic and the LCD.

Assuming a continuous measurement, we have the following currents:

130 µA	Ceramic oscillator	if on continuously
150 µA	Bipolar comparator	if on continuously
< 25 µA	Digital part incl. TDC	average value

Tab. 3: Further current drains, all data for 3 V

The sum of those currents is about 300 µA. Adding the current required by the load capacitor and the LCD, the total demand should be in the 300-400 µA range. This is a drastic reduction from the current demand of the classical solutions, but still too much for making a solar scale.

We must further reduce the current by selecting the correct sequence control. Simply put, we must configure the device in such a way that all parts are active only when they are required for the measurement. This helps significantly reduce the current consumption of the oscillator, the comparator and the digital logic.

Looking at the example above with 50 discharge cycles per second and a cycle time (sum of discharge and recharge time) of 100 µs, the total measurement time is:
50 discharges x 100 µs cycle time = 5 ms
+ additional 100 µs for the setup time of the oscillator, in total: 5.1 ms

The active time is 5.1 ms / 1000 ms = 0.0051 = 0.51% (1 measurement/ s)
This means that the oscillator current can be reduced to 130 µA x 0.0051 = 0.663 µA (!).
An additional 30% of the comparator current can be recovered because it is needed only for the duration of the discharge cycle. In this example the active period is only 0.0051 x 0.7 = 0.00357, so the comparator current is only 150 µA x 0.00357 = 0.536 µA (!). The current of the digital logic can also be reduced to only 2 to 3 µA due to the reduced conversion time and processing time.

1.0 Preliminary Considerations

We still have to consider the current consumption of the LCD. This current cannot be assigned directly to the PSØ8 chip, but it has to be taken into account in the total current calculation. In general, larger active LCD panels lead to higher current demands. Physically, the current depends on the capacitive reloading of the segments and the resistance of the liquid crystals which decreases as segment size increases. This reveals that the size of the displays digits is the main reason for the increase in current demand. Depending on the size of the digits, the LCD current may vary from 1 to 30 µA.

The PSØ8 has an integrated LCD driver that can drive displays from 2.0 V to supply voltage. Current can be saved when the internal voltage doubler with charge pump is not used, but instead the LCD is driven directly from the supply voltage (Direct Drive Mode). This is no problem for solar scales because the PSØ8 needs a regulated voltage in this case. Selecting an appropriate display allows the system to reach a minimum LCD current of about 2 µA.

The previous theoretical considerations did not take into account the resolution (to be controlled by averaging) and were all done for 1 measurement per second (1 Hz). Increasing the update rate to e.g. 3 Hz mainly affects the load capacitor current, the comparator and the oscillator. The summary in our example is:

Charging capacitor	7,5 µA	x 3	22,5 µA
Oscillator (active time only)	0,66 µA	x 3	1,98 µA
Comparator (active time only)	0,53 µA	x 3	1,59 µA
Digital part (average)	2,5 µA	x 1	2,5 µA
LCD-Display (small digits)	2,5 µA	x 1	2,5 µA
Total current at 3 Hz	16,69 µA		**31,07 µA**

Tab. 4: Example total current

In addition to the correct settings for the PICOSTRAIN measurement, resolution and update rate settings are very important. The current consumption increases with higher resolution and higher update rate. On the other hand, weigh scales with low resolution and update rate like body scales can be built with very low current consumption.

1.0 Preliminary Considerations

Please note: as we will see during the realization of a solar scale in section 2, it is possible to reduce update rate and resolution to have a minimum current consumption. But then the number of measuring points is very low. This might result in undersampling.

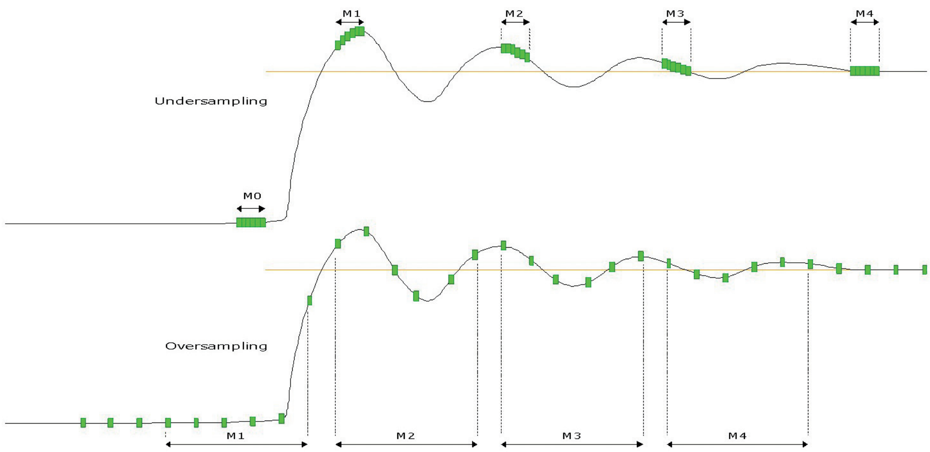


Fig. 7: Undersampling and Oversampling

The Shannon and Nyquist theorems state that a sufficient frequency of measurement data (at least two times the signal frequency) is needed to cover and reconstruct the original signal completely [5]. The above figure shows that in case of undersampling the measurement value is not captured precisely. The information between measuring points M1 and M4 is missing. The display is unstable with varying load due to vibrations (e.g. caused by a passing truck). With oversampling the signal is sampled with sufficient frequency and can be measured correctly.

There are applications where undersampling is not relevant. In body scales e.g. the intrinsic mechanical oscillation frequency is much too low. Other applications such as postal scales will be sensitive to undersampling. Here another feature, the Stretched Mode, of the PSØ8 will help.

1.0 Preliminary Considerations

Stretched mode combines the advantage of few measurements (saves current) and a reasonable distribution of these measurements for avoiding undersampling. Figure 6 showed that in single conversion mode it is possible to set a period without measurements, with the risk that the sampling frequency is too low. Stretched mode addresses this risk in the following manner:

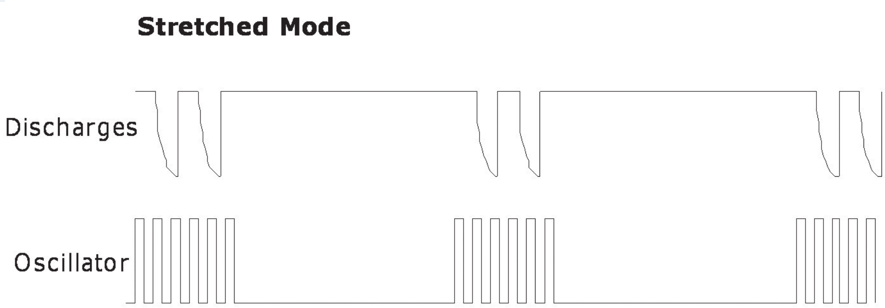


Fig. 8: Stretched mode "stretches" measurements to avoid undersampling

The discharge cycles are stretched in a way that the total number is not increased but the distribution is improved. The advantage is to have sufficient sampling points for a correct measurement. The oscillator and comparator are switched on again only for the duration of the measurement. This keeps the current saving effect. The oscillator current is slightly increased because the oscillator needs settling time which affects the total current calculation. Considering the 50 discharge measurements assumed above calculations yield the following:

50 x 100 µs = 5 ms –> 130 µA total comparator current x 0.005 = 0.65 µA.
This new technology is unique and unparalleled in the world of classical A/D converters.

Conclusion: It is necessary to have a close look at the correct sampling rate for weigh scales. In applications that target least current consumption and still need a correct oversampling the PS08 stretched mode is the right choice.

1.4 Case: Total Darkness

In section 1.2 it was shown that the right choice of the solar panel under the given lighting conditions is essential for the proper operation of the weigh scale. But what happens if there is no light at all? This happens each day for several hours. Is it possible to measure even in this situation, and if so, how?

The short answer is that in case of short periods of missing light, seconds to a few minutes, measuring is still possible. But when the weigh scale is in darkness for a longer time, then measuring will be possible only after a period of several seconds with sufficient light. In this case it is necessary to ensure the correct power-up of the weigh scale.

Some issues have to be taken into account to guarantee correct behavior:

a) A voltage regulator and a buffer capacitor for intermediate charge storage are necessary.

b) A voltage detection for the correct switch-on level has to be implemented.

c) An external power-up circuit is necessary to ensure a correct power-up behavior even when starting from lowest voltages.

The first requirement (a), implementation of a buffer capacitor, offers several advantages. This buffer capacitor stores charge and can provide it when needed. This buffer capacitor is placed directly after the solar panel, ahead of the power-up circuit. With sufficient light it is charged to the voltage of the solar panel. It buffers variations in light and therefore current. It may provide the needed current when there is not sufficient light for a short time. In other words, it acts like a short-time accumulator.

1.0 Preliminary Considerations

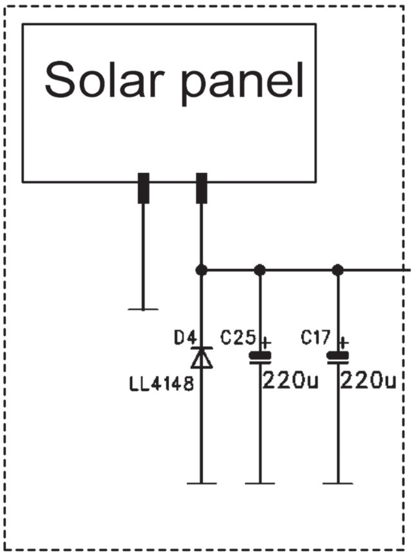


Fig. 9: Buffer capacitor in between solar panel and power-up circuit

Example: We use an electrolytic capacitor of 440 µF for buffering. If its voltage is 1 V above the output voltage of the voltage regulator and the average current of the weigh scale is 14 µA then this capacitor may buffer for 32 seconds. In standby-mode no measurements are done and the current of the weigh scale is far less than 10 µA. In this case the capacitor can buffer for minutes.

By adhering to the second requirement (b), the voltage detection, it is possible to define levels for switching off the scale and switching it on again. There will be a point as the voltage drops where no further measurements are reasonable. Alternatively, before switching on the scale, it is necessary to have a stable and sufficient voltage level and sufficient current. The voltage itself is a good indicator to decide when the scale can be switched on and measurements can be started. This way we avoid allowing the system to start below a dedicated level and run into an unstable state.

For this reason, the PSØ8 has an integrated voltage measurement. It starts at 2.1 V and provides the result directly to the microprocessor program. Various actions can be programmed according to different voltage levels. In case of total darkness the voltage will drop below the 2.1 V level. Therefore, it is necessary to have an additional external circuit to cover this case.

1.0 Preliminary Considerations

An effective solution is discussed in requirement (c), the inclusion of an external power-up logic. This small additional circuit is described in section 2.1 and guarantees the proper start-up operation even after long-lasting darkness. In principle the circuit checks whether the voltage is stable for a specific time before switching on the voltage for the rest of the system.

It is possible to have a secure switch-on behavior by adding measures (a) and (c). During short interruptions of light the system can still measure. Total darkness is a special task and has to be taken into account while designing the hardware and software.

2.0 Realization of a solar scale

2.1 Design of the Scale

In this second section we show step-by-step how to build a solar scale. All considerations assume a body scale with 4 half-bridge sensors of 1 kΩ strain gauges each. The maximum load is 150 kg and shall be resolved by 0.1 kg (1500 stable scale divisions). The scale shall be made without a battery. The power is supplied only by two solar panels of type SS-5520 from Sinonar. The current consumption shall not exceed 20 µA during the measurement (= during weighing).

The following design considerations are of basic nature for quattro scales and may be used as reference design. In section 2.4 we then show the real performance data of such a scale.

Design: A comparison of a conventional A/D converter solution and a PICOSTRAIN solution shows that the sensor setup itself is simplified by the capability of measuring half-bridges (no need to combine them to full-bridges). Also the integrated microprocessor saves components and design efforts.

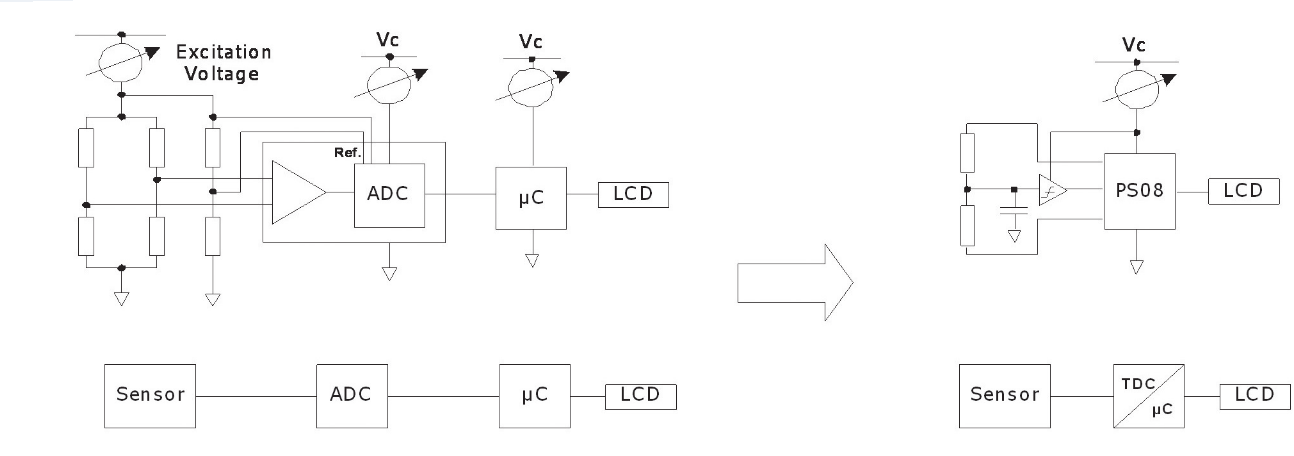


Fig. 10: Simplified design with PSØ8 compared to an A/D-converter solutions

2.0 Realization of a solar scale

How does a PICOSTRAIN design with PS08 look like?

The following block diagram shows the main elements for a solar scale designed with PS08:

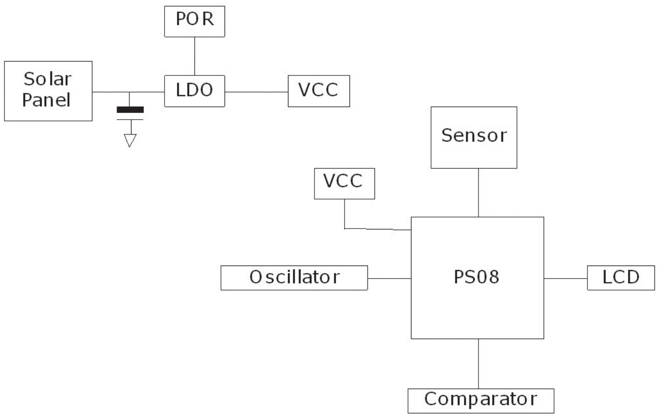


Fig. 11: Block diagram solar scale with PS08

It is obvious from the block diagram that the PS08 itself needs only a few external components, e.g. the comparator and oscillator. Some effort has to be spent on the power supply to make it stable and proven. Below we look into some details of the schematics, mainly the power supply and the specialties of PICOSTRAIN. A complete schematic is given in the appendix of this document.

Hints for the power supply:

a) Use a solar panel that is big enough
b) Use a big buffer capacitor, e.g. 440 µF
c) Use a low-current linear regulator (LDO), 3V type
d) Use a low-current power-up regulator (named POR in the block diagram)
e) Decouple the various voltages by low-pass filters

2.0 Realization of a solar scale

Hints for PICOSTRAIN:

f) Use a ceramic resonator as oscillator

g) Build the comparator with 6 transistors according to the recommended schematics

h) Connect the sensors directly to the PSØ8

i) Select size of C_{load} according to our recommendations

j) Use blocking capacitors for PSØ8

The size of our solar panels, Sinonar SS-5520, is 5.5 cm x 2 cm = 11 cm². According to the table in section 1.2, they deliver about 14 µA at 200 lux (weak office light). We calculated the maximum current demand of the system to be about 20 µA. Therefore, we use two SS-5520 panels. Alternatively the SS-6728-A panel, roughly 19 cm² in size and 25 µA of operating current, might be used.

For the buffer capacitor we recommend an electrolytic capacitor of 440 µF. This will be sufficient to buffer a few minutes at the estimated average current consumption.

Having a linear regulator is absolutely necessary for a stable supply voltage. We need a low-current type and therefore select the XC6206P30 from Torex Semiconductor.
Important: Do not use switching regulators!

The power-up logic is a small additional circuit following the LDO. The task of this circuit is to switch on the power to the PSØ8 correctly after a period of total darkness.

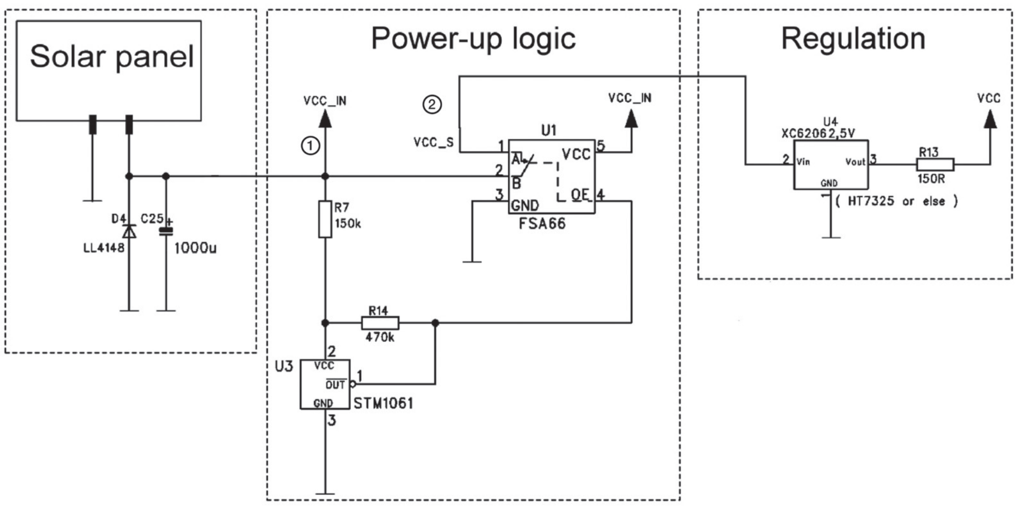


Fig. 12: Power-up logic in detail

2.0 Realization of a solar scale

The circuit is made of the reset device STM1061 from ST-Microelectronics (low current), the analog switch FSA66 from Fairchild Semiconductor as well as some resistors and a capacitor. The functionality follows:
The voltage at the buffer capacitors is checked at point 1 by the reset circuit. In case this voltage is stable for a while at a level of 3 V, the reset device drives the analog switch to pass the voltage towards point 2. So the voltage comes to the LDO and is regulated to Vcc = 2.5 V. This voltage is supplied as Vcc to the rest of the circuit.

The PSØ8 needs several supply voltages for the various parts of the chip. Those voltages can be generated from one source, but should be decoupled by low-pass filters (RC networks). For short times, high currents will be needed due to the pulsed operating principle of PICOSTRAIN. This might cause cross-talk which can be reduced by the decoupling.

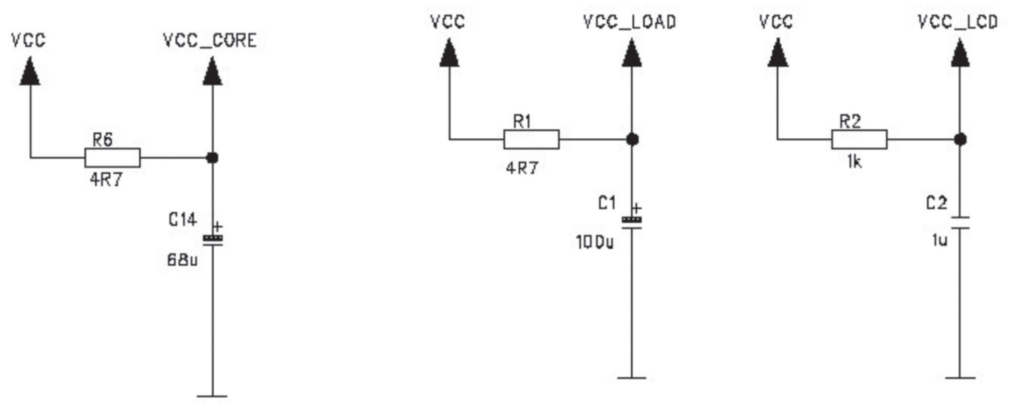


Fig. 13: Decoupling of the voltages by low-pass filters

In the next section we give some PICOSTRAIN-specific hints.

For the oscillator we recommend a simple ceramic resonator. The 4MHz CSTRCG4-MOO-G53A from Murata should be sufficient. This ceramic resonator has a very short settling time and the precision is sufficient. We recommend against quartz oscillators because of their long frequency settling time.

2.0 Realization of a solar scale

The comparator defines the trigger level for the time interval measurement and thus has a direct influence on the measurement quality. Over the years we developed a comparator circuit that shows excellent results with respect to noise and low current consumption.

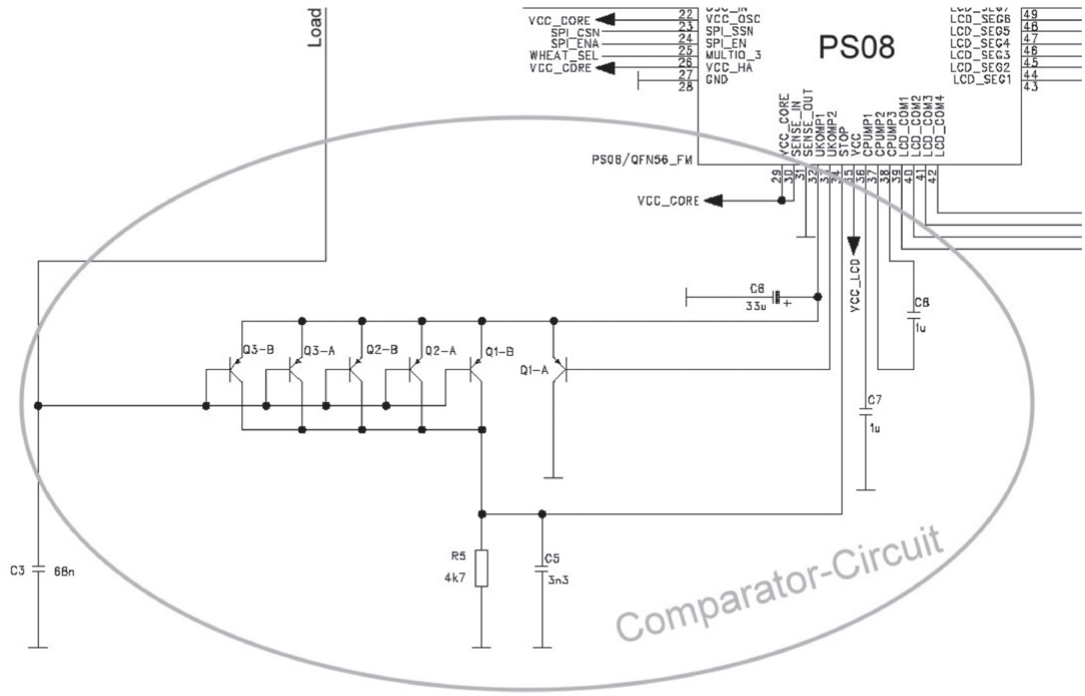


Fig. 14: Recommended comparator circuit

Basically, the comparator is made of 6 transistors, 2 capacitors and 1 resistor. Putting 5 transistors in parallel minimizes the noise. It is not necessary to have matched transistors. It is possible to use standard PNP transistors. They are available as two-in-a-package PNP transistors. The CMKT5078 from Central Semiconductor is a good example and is offered in the very small SOT-363 package. The values for R5, C5 and C8 can be taken from the schematics.

As an alternative to the external comparator it is possible to use the PSØ8 internal comparator. The external circuitry will be reduced, but the noise will be higher and the resolution reduced by about 0.6 bit. Sample circuits are shown in the PSØ8 data sheet, available at www.acam.de.

2.0 Realization of a solar scale

With PICOSTRAIN the sensors are connected directly to the chip. Due to the pulsed operation they are powered only during the measurement. The connection of 4 half-bridges is shown in the next figure:

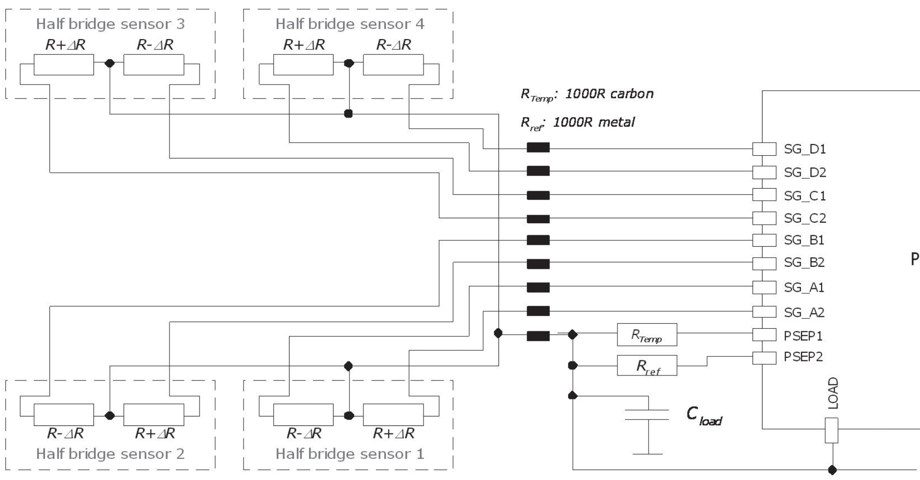


Fig. 15: Connecting the strain gauges to PSØ8

The 4 half-bridges are all connected directly to PSØ8, the center tap as common line to the load capacitor. In the sample scale the wiring of the 4 half-bridges is quite long forcing us to consider the effects of EMI. The 50 Hz signal at main voltage is the most likely EMI aggressor to cause trouble. Higher frequencies coming from devices such as mobile phones are less critical due to the PICOSTRAIN method. **In order to reduce the EMI sensitivity, we strongly recommend that the 3 wires of each half-bridge are twisted. Further, it is mandatory that each half-bridge load cell is grounded (connected to the PCB's GND).**

The last issue is the selection of the load capacitor. The size and quality of the load capacitor directly affects the measurement. The current into this capacitor is the same as the current into the strain gauges. We recommend for Cload 68nF, X7R.

The discharge time is generally calculated as: t = 0.7*R*C. In our case the discharge time is 0.7 x 1 kΩ x 100 nF = 70 µs. In general the discharge time should be in the range of 30 µs to 150 µs. Under the aspect of a current saving solar application the value should be close to the lower limit.

2.0 Realization of a solar scale

In the following section, we review some guidelines for blocking capacitors. They may be standard 100 nF ceramic capacitors. They should be placed very close to the PSØ8.

Layout Hints:
Follow these guidelines so that the design recommendations from above can be implemented most effectively.

- Place the oscillator very close to the PSØ8
- Do not cross the SPI lines with the load or port lines
- Do not cross the load line with the oscillator lines
- Place the electrolytic capacitor of VCC-Load as close to pin 16 as possible
- Keep the port lines as short and symmetric as possible
- If possible use flooded planes around the oscillator

In general, the layout has a big influence on the measurement quality, especially for 2-layer boards. Over the years we have collected a lot of useful experience that is now represented in the layout of our evaluation system. The schematics and layout data are available from acam for free. They give a straight indication for a good design with PICOSTRAIN.

2.2 Programming the Device

The integrated, proprietary microprocessor of the PSØ8 allows compact and time efficient programming in assembly language. Due to the strong interaction of the processor and converter, the program is closely adopted to the structure of the hardware. The following diagram shows a measurement sequence and the microprocessor calls:

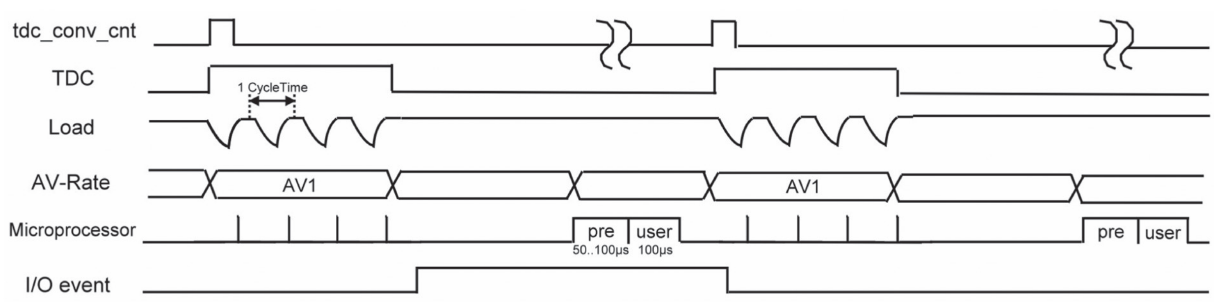


Fig. 16: Measuring sequence and processor calls in PSØ8

2.0 Realization of a solar scale

The so-called single conversion counter (set by tdc_conv_cnt) defines the time frame, the time window for a complete measurement. Within this time frame a number of discharge time measurements are united, 2 discharge time measurements for each half-bridge. With avrate=1 a total of 8 discharge time measurements will be made.
The microprocessor is switched on and coordinated by the TDC. This is done for a short time after each discharge time measurement, but mainly at the end of the whole measurement. Then, the microprocessor does the data pre-processing and afterwards starts the user program. After that, a new measurement may start and the sequence starts again from the beginning.

From a programmers point of view it is important to understand that the microprocessor and converter act in the interdependent manner shown below.

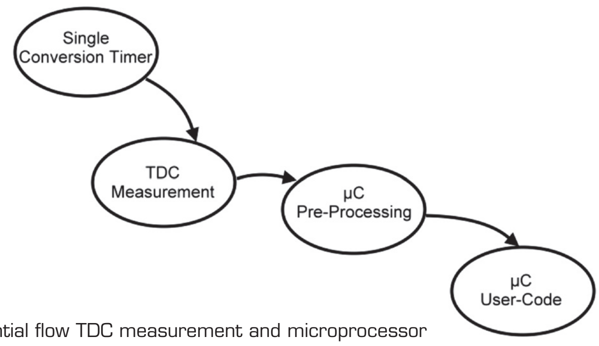


Fig. 17: Sequential flow TDC measurement and microprocessor

This structure of the PSØ8 can be found again in the structure of the program. It is necessary to write the program state controlled, because between each call of the user program there is a measurement. This means, that a state variable defines the last states of the program and which state causes the the PSØ8 to continue with the program.

Example: The intention is to get the offset of the weigh scale. Therefore we make 10 measurements without load and calculate the average. This average is subtracted from the following measurement results.

The methodology used to acquire the offset follows. We make 10 measurements and therefore jump into the program 10 times. The program has to recognize the ‚Get-offset' state. A counter counts 10 measurements. After the 10th measurement the average is calculated and the next state, a Measuring-state, is set. In other words, **each loop is not just a program loop but also includes a TDC measurement!**

2.0 Realization of a solar scale

The following hints may help to understand how to build a program:

- Programming by states helps to structure the program
- The reason for the jump into the user code should be detected (the processor offers flags in the status bytes)
- Setting additional flags in the RAM allows the state to be saved between several program passes

Based on these thoughts and the discussion from section 1.4 (total darkness) we arrive at the following basic structure of the program:

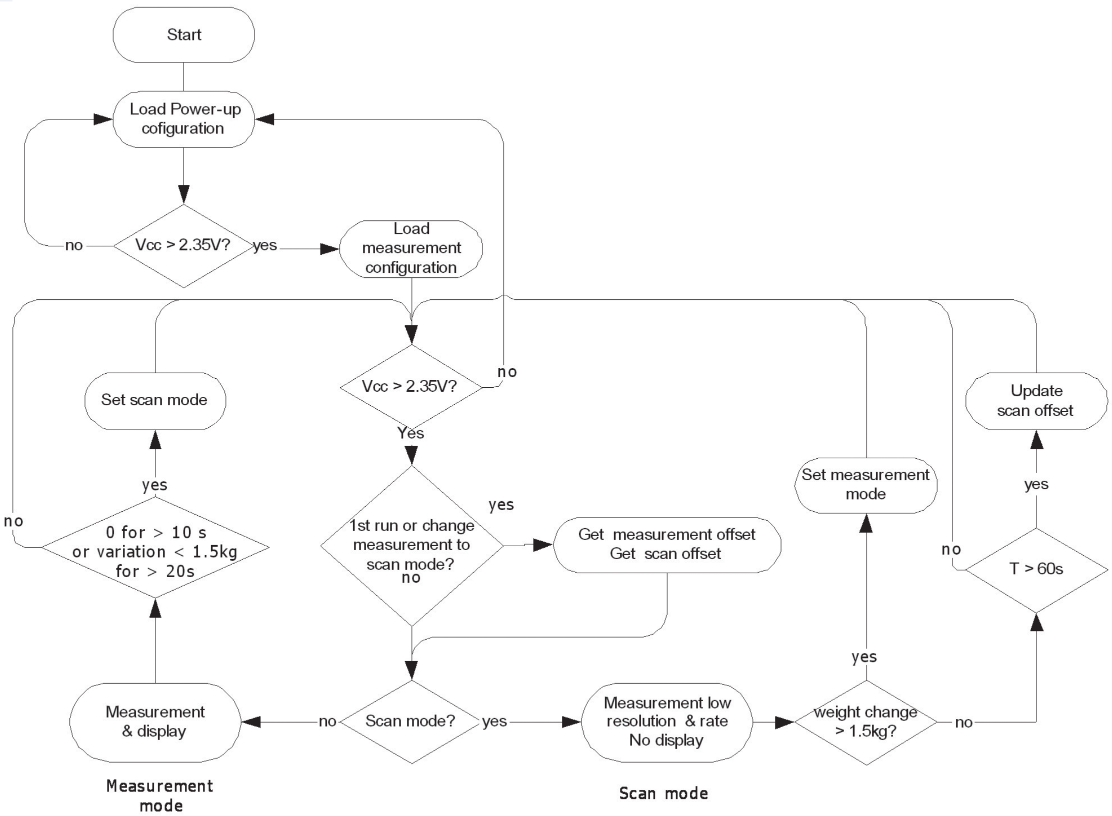


Fig. 18: Principal structure of the program

The flow diagram shows 3 principle states or configurations:

Power-Up state - minimum current consumption to power-up the system
- LCD, comparator, oscillator, etc. are switched off
- no measurement
- minimized current consumption

2.0 Realization of a solar scale

Measuring state - active measurement, LCD, comparator, etc. are active
- display result on the LCD
- measuring with averaging 2 and 1.5 Hz update rate

Scan mode - active measurement, but lower update rate
- no display on the LCD
- measuring with averaging 2 and 0.8 Hz update rate

By means of those 3 configurations it is possible to operate the system reliable and with a very low total current consumption.

The program flow is the following: Initially the program starts with a special configuration (‚power-up-config'). This one is characterized by the lowest possible current consumption and checks the voltage on a specific level. In case this level is reached and stable for a minimum 5 seconds, a configuration for the measuring mode is loaded. Afterwards the offset is measured and then the scan-mode configuration is loaded (for saving current). In scan mode the parameter setting is different and therefore the offset differs. It is necessary to measure the offset value in scan mode separately to detect a load jump of 1.5 kg correctly. After getting this second offset value, the chip is in scan mode and makes only a few measurements at low resolution and reduced update rate. The LCD is switched off. As soon as a load jump of 1.5 kg is detected the program switches into the measuring state. The result of the measurement is immediately displayed on the LCD.

Special case voltage detection: In section 1.4b we explained that the voltage measurement helps to check whether voltage and current are sufficient to run a measurement or not.

Special case initial offset: The initial offset is subtracted from later measurement results to make sure that the display is 0.0 kg at zero load. This zero-offset will drift over temperature and time. Therefore it is reasonable to track the offset. It might happen that there is a load change during the offset calculation. Then, the offset calculation is canceled and the program jumps immediately into the measuring state (blue arrow figure 18).

This program sequence is realized in assembler. The complete program is given in appendix A2.

2.0 Realization of a solar scale

The next section gives a close look into the 3 configurations and explains the parameters in detail.

2.3 Configuration of the Device

The former section revealed that we need several different configurations to ideally control the current consumption. The 3 configurations are:

	Power-Up	Measuring	Scan
Register 0:	0x64820A	**0x64828A**	**0x1E828A**
Register 1:	0xAFC803	**0xADC003**	0xAFC803
Register 2:	0x008356	**0x0084DE**	**0x0084D6**
Register 3:	0x000210	**0x040217**	**0x040210**
Register 4:	0x7FFFFF	0x7FFFFF	0x7FFFFF
Register 5:	0x7FFFFF	0x7FFFFF	0x7FFFFF
Register 6:	0x7FFFFF	0x7FFFFF	0x7FFFFF
Register 7:	0x7FFFFF	0x7FFFFF	0x7FFFFF
Register 8:	0x0F1687	0x0F1687	0x0F1687
Register 9:	0xFFC568	0xFFC568	0xFFC568
Register 10:	0x2D008F	0x2D008F	0x2D008F
Register 11:	0x2A18C0	**0x2A08C1**	0x2A18C0
Register 12:	0xE8E728	0xE8E728	0xE8E728
Register 13:	0x880040	0x880040	0x880040
Register 14:	0x57ABD5	0x57ABD5	0x57ABD5
Register 15:	0x000000	0x000000	0x000000
Important Settings:	AVRATE = 2 CYTIME = 53 TDC_CONV_CNT = 100 TDC_SLEEPMODE = 1 BRIDGE = 0 PPTEMP = 0 LCD_STANDBY = 1 CON_COMP = 1 SEL_START_OSZ = 0 MFAKE = 0	AVRATE = 2 CYTIME = 77 TDC_CONV_CNT = 100 TDC_SLEEPMODE = 0 BRIDGE = 3 PPTEMP = 1 LCD_STANDBY = 0 CON_COMP = 1 SEL_START_OSZ = 2 MFAKE = 1	AVRATE = 2 CYTIME = 30 TDC_CONV_CNT = 30 TDC_SLEEPMODE = 0 BRIDGE = 0 PPTEMP = 0 LCD_STANDBY = 1 CON_COMP = 1 SEL_START_OSZ = 2 MFAKE = 1
Current:	~ 1-2 µA	~ 14 µA	~ 5 µA

Tab. 5: Overview of the 3 configurations

2.0 Realization of a solar scale

Important general settings for all configurations are:
DIS_OSC_STARTUP = 1
EPR_PWR_CFG = 0
SEL_COMPR = 2

The following table gives a short description of the main parameters and shows the registers where they are found (alphanumerical order):

Parameter:	Register, Bits:	Explanation:
AVRATE[9:0]	**Register 2, [23:14]**	**Number of internal averaging**
BRIDGE[1:0]	**Register 3, [1:0]**	**Number of half-bridges** **1 = 2 half-bridges** **3 = 4 half-bridges**
CON_COMP[1:0]	Register 11, [1:0]	Comparator setting 0 = off 1 = on only during measurement
CYTIME[7:0]	Register 2, [13:4]	Sets the cycle time (1 charge + discharge cycle) in multiples of 2 µs if stretched-mode is off, in multiples of 100 µs if stretched-mode is on. Here: CYTIME = 77 makes 154 µs
DIS_OSC_STARTUP	Register 0, [3]	Minimize oscillator current when switching it on
EPR_PWR_CFG	Register 1, [2]	Defines whether the configuration is loaded from the EEPROM after a reset. 0 = is not loaded 1 = is loaded
LCD_STANDBY	Register 11, [12]	Sets the LCD into standby 0 = LCD on 1 = LCD off (Standby)
MFAKE[1:0]	Register 3, [3:2]	Sets the number of fake measurements. These make the measurement more stable but cost current
PPTEMP	Register 2, [3]	Activates the gain-error correction respectively the temperature measurement 0 = disabled 1 = enabled
SEL_COMPR [1:0]	Register 0, [15:14]	Sets the operating resistance of the comparator 2 = 7 kOhm
SEL_START_OSZ [2:0]	Register 3, [19:17]	Sets the delay between switching on the oscillator and starting the measurement (settling time). 0 switches off the oscillator. 0 = Oscillator off 2 = 100µs Delay
TDC_CONV_CNT [7:0]	Register 0, [23:16]	Sets the single-conversion timer, this means the time to the next measurement cycle, in multiples of 6.4 ms. E.g. a setting of = 100 gives a conversion time of 640 ms –> 1.56 Hz.
TDC_SLEEPMODE	Register 1, [17]	Switches off the TDC, no strain measurement. Is used e.g. if only buttons shall be scanned. 0 = TDC is active 1 = TDC is not active (Sleep mode)

Tab. 6: Explanations of important parameters

2.0 Realization of a solar scale

The table explains the most important parameters with regards to solar applications. Other parameters are explained in the PSØ8 datasheet, available from www.acam.de.

The three configurations define the three operating modes. Power-up state, for example, is used to start up with lowest possible current. It is used when starting from total darkness or in case the light is not sufficient.

The measurement state was configured to reach the requested resolution and update rate. Most of the PSØ8 functional units are active as well as the comparator, oscillator and LCD. With the optimized configuration the total system current in the measuring state is reduced to approximately 14 µA.

In scan mode the measurements are done in longer intervals and the result is not displayed on the LCD. Every 60 s a new offset is calculated to avoid a zero point drift caused by the drift of the sensors. In this configuration the current is reduced to 5 µA.

Depending on the program state, the configurations are loaded dynamically. The most important transition conditions have been explained in section 2.2 already.

2.4 Realized Example

Based on the general considerations for making a solar scale and the dedicated investigations on how to implement them into a design, acam built its own test setup for a solar scale. Therefore we took a quattro body scale off the shelf and replaced the electronic board with our own board.

We used two SS5520 solar modules from Sinonar to provide power. For the display we used an LCD with an average current consumption of 8 µA.

2.0 Realization of a solar scale

Further specifications coincide with the example provided earlier of a body scale with a maximum load of 150 kg and 0.1 kg resolution (1500 scale divisions), 4 half-bridges with 1 kΩ strain gauges and an update rate of 15 Hz.

Measuring mode:	**Scan mode:**
AV-Rate = 2	AV-Rate = 2
Update rate = 1.5 Hz	Update rate = 0.8 Hz
Resolution = 1500 division stable	
–> ~ 14 µA current consumption	–> ~ 5 µA current consumption

The main parameters are:
The target resolution of 1500 stable divisions was reached with an average current of about 14 µA (7 µA consumed by the LCD). After measuring, the scale automatically switches into scan mode which needs only 5 µA. In scan mode a blinking sun on the LCD indicates that the scale is ready for measurement. The offset is corrected every 60 seconds. In case the load changes by more than 1.5 kg the scale automatically switches into measuring mode. The weight is displayed immediately. In this version of the program the display was not stabilized nor frozen as is frequently done in body scales.

Summary

At the very beginning of this document we raised a question about the possibility of building a pure solar scale with strain gauges. This white paper showed in detail, that there is a solution provided by PICOSTRAIN technology that resolves the shortcomings of the classical A/D converter approach. PICOSTRAIN by principle consumes very little power.

The white paper showed step-by-step what has to be considered when designing a solar scale and how this can be implemented in the design. The practical realization of a body scale in a test setup with an average current consumption of 14 µA in measuring mode proves the validity of the theoretical assumptions.

Outlook

By means of the realized solar driven body scale we saw the capability and potential of the PICOSTRAIN measuring principle. From our point of view, the advantages of PICOSTRAIN and especially the solar capability may be transferred to other types of scales and other strain gauge applications.

In the consumer market these could be postal scales, pocket scales or kitchen scales. With bigger solar panels or good lighting conditions the available current might reach 100 µA. In this case even industrial scales might be an option.

- Legal-for-trade scales class II following EN45501 up to 3,000 divisions

- Well-performing industrial scales with up to 20,000 stable divisions

With PICOSTRAIN it is finally just a question of the available current whether such scales can be made. Furthermore, there are many more strain gauge based applications like force or pressure sensors which can benefit from the advantages of PICOSTRAIN. Also wireless sensors open an interesting field as current consumption is an issue there, too.

We greatly appreciate your interest in our technology. Solar driven devices are not only elegant and come with many advantages, but they also show the manufacturer's capacity for innovation. We thank you for your interest and look forward to further communication.

The acam team.

Bibliography

[1] http://www.sinonar.com.tw
[2] http://de.wikipedia.org/wiki/Solarpanel
[3] http://www.oeko-energie.de/solarmodule.htm#Solarzellen_
[4] http://www.schott-solar.de/de/unsereprodukte/oemprodukte/oemindoor.html
[5] http://de.wikipedia.org/wiki/Unterabtastung

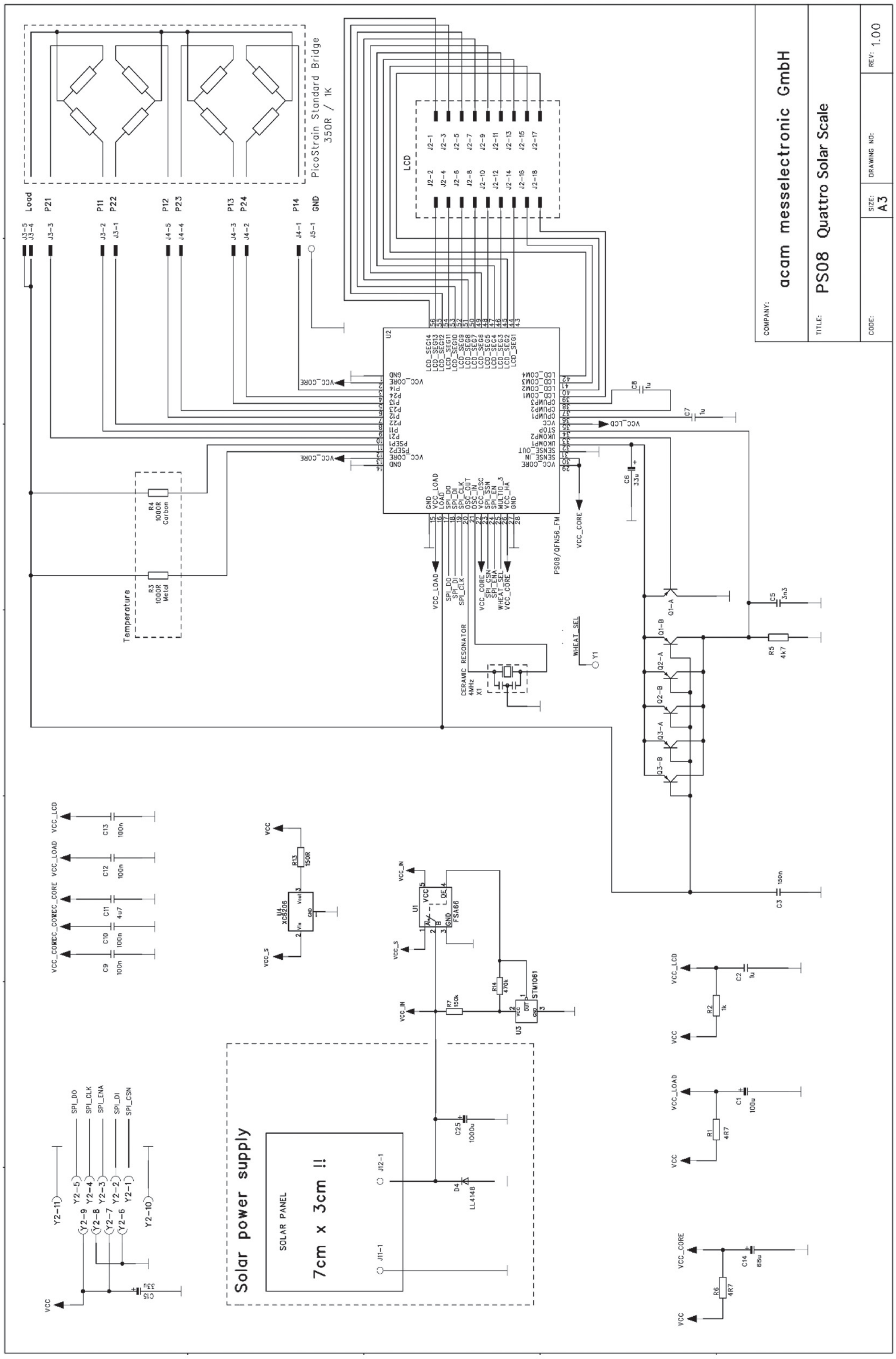
Additional Links

www.acam.de Official website of acam-messelectronic gmbh

Appendix

A1 Schematics solar quattro scale
A2 Assembler program for solar quattro scale

Appendix A1: Schematics Solar Quattro Scale



Appendix A2: Assembler Program for Solar Quattro Scale

```
<COMMENT>
;-------------------------------------------------------------------------------
;           Software for Quattro weight scale with solar cells as only power ssupply (no battery backup)
;           File: Quat_solar_eigen_newPOR_14.asm
;           Author: UTG
;           Date: 11-09-08
;
;           Description: Complete scale program for solar application with appr. 17µA current  consumption when ON at a resolution of 100,0g.
;           In the OFF state current consumption appr. 3µA (with LCD). Scale turns on automatically when weight is applied (min 2000gr).
;
;           Program features include:
;           - 5 time rolling average
;           - zero offset
;           - offset adjustment for creeping
;           - LCD display for 5 digits
;           - 2 unit modes: kilogramm and pound with 1 digit after the comma
;           - auto-OFF
;           - auto-ON
;           - Scan offset in Off mode to display correct weight immediately after weight change detection (turn ON)
;           - In scan mode offset is taken appr. every 60sec.
;
;           Current functionality:
;           On reset or power returning >2.9V init offset is performed and scale measures. Turns off after a (fairly short) time at 0kg(+/-50g).
;           When OFF scale runs in scan mode with slower measure rate and no pptemp; every 60s. init offset for ON mode is calculated.
;           On AUTO-ON, jump of weigth > 3000g, the init offset which was taken every 60s is used and scale measures until auto-off.
;           Current consumption in ON mode appr. 24µA avg. ; in OFF mode appr. 15µA (in this version the LCD stays on which is about 9µA)
;
;;-------------------------------------------------------------------------------
;           # auto_on at:                                                    3 kg
;           # auto_off at zero after                                         app. 10sec.
;           # auto_off at stable measurement                                 app. 20sec.
;           # threshold for value change to reset auto-off counter           1,5kg
;           # hysteresis for display measurement                             100gr
;           # hysteresis at zero                                             400gr
;           # overload at                                                    150kg
;           # low voltage turn off at                                        2,35V
;           # threshold for value change detection to abort taking offset    1,5kg
;           # threshold for scan offset value to be considered stable        500g
;           # threshold for measure offset value to be considered stable     67,5g
;           # threshold for value change detection to re initialize the filter 500g
;           # Number of values taken for init offset:                        6
;           # Take new measurement offset in scan every:                     60 sec.
;           # threshold for discarding new offset value while in scan        1,5kg
;-------------------------------------------------------------------------------
;
;           New in v18:  - Display of OFSEt when offset is taken
;           - Changed threshold for discarding offset during scan to correct value (app. 1.5kg)
;
;
<ENDCOMMENT>
equal  0x64820A            ; Config. reg. 0
equal  0xAFC807            ; Config. reg. 1
equal  0x001356            ; Config. reg. 2           AVRate 0, cycle time 77, no pptemp
equal  0x000215            ; Config. reg. 3           ; 4 MHz OFF    1 MFake, 2xHB
equal  0x7FFFFF            ; Config. reg. 4
equal  0x7FFFFF            ; Config. reg. 5
equal  0x7FFFFF            ; Config. reg. 6
equal  0x7FFFFF            ; Config. reg. 7
equal 0x0F1687             ; Config. reg. 8
equal 0xFFC568             ; Config. reg. 9
equal  0x2D008F            ; Config. reg. 10
equal  0x2A18C0            ; Config. reg. 11                   Comp. OFF
equal 0xE8E728             ; Config. reg. 12
equal  0x880040            ; Config. reg. 13          ; LCD direct drive
;equal  0x7FFFFF           ; Config. reg. 14
equal  0x57ABD5            ; Config. reg. 14
equal  0x000000            ; Config. reg. 15

#include "get_button_edges_into_status.h"

;ATTENTION:          the programme works only when programming plug is disconnected after download --> stand alone application
;--------------------------------------------------------------------------------------------------------------
   CONST flg_mio_io3              23        ; Pin29 gedrueckt
   CONST flg_sck_io2              22        ; Pin22 gedrueckt
   CONST flg_sdi_io1              21        ; Pin21 gedrueckt
   CONST flg_sdo_io0              20        ; Pin20 gedrueckt
   CONST flg_rstpwr               19        ; EEpromeinsprung durch Poweron Reset
   CONST flg_rstssn               18        ; EEpromeinsprung durch Tastendruck an
   CONST flg_wdtalt               17        ; EEpromeinsprung durch Watchdog-Interrupt
   CONST flg_endavg               16        ; EEpromeinsprung durch fertige Messung
   CONST flg_asleep               15        ; EEpromeinsprung im Sleep-Modus
   CONST flg_ub_low               14        ; Batteriespannung unterschritten
   CONST flg_errtdc               13        ; Fehler im TDC
   CONST flg_pslock1              12        ; Lock des phase shifters
   CONST flg_pslock0              11        ; Lock des phase shifters
   CONST flg_errprt               10        ; Fehler an den DMS-Ports
   CONST flg_timout               09        ; Timeout TDC
                                            ; not used
   CONST flg_mio_r_io3            07        ; steigende Flanke an Pin29
   CONST flg_sck_r_io2            06        ; steigende Flanke an Pin22
   CONST flg_sdi_r_io1            05        ; steigende Flanke an Pin21
   CONST flg_sdo_r_io0            04        ; steigende Flanke an Pin20
   CONST flg_mio_f_io3            03        ; fallende  Flanke an Pin29
   CONST flg_sck_f_io2            02        ; fallende  Flanke an Pin22
   CONST flg_sdi_f_io1            01        ; fallende  Flanke an Pin21
   CONST flg_sdo_f_io0            00        ; fallende  Flanke an Pin20
```

```
;-------------------------------------------------------------------------------------------------------
          CONST first_turn_off_flag                 36
          CONST turn_off_flag                       35
          CONST ram_track_offset_cnt                34
          CONST ram_unstable_counter                33
          CONST ram_turn_on_delay                   32

          CONST ram_tmp_unit_digit                  19
          CONST     ram_tmp_mlt_hb0                 17
          CONST ram_tmp_unit_factor                 16
          CONST roll_avg_mean_5                     15
          CONST state_ram_tmp_cnt                   14
;-------------------------------------------------------------------------------------------------------
          CONST state_ram                           13
;-------------------------------------------------------------------------------------------------------
          ; content of state_ram byte:
          CONST state_bit_off                       0
          CONST state_bit_init_offset               1
          CONST state_bit_gauge                     2
          CONST state_bit_offset_auto_off           3

          CONST state_bit_init_offset_off           5
          CONST state_bit_low_volt_off              6
          CONST state_bit_auto_on_now               7
          CONST state_bit_first_run                 8
          CONST state_bit_startup_failed            9
;-------------------------------------------------------------------------------------------------------

          CONST ram_last_value                      12
          CONST ram_initial_offset                  11
          CONST     ram_last_display                10

          CONST ram_seg_run_value                   9
          CONST ram_scale_unit_mode                 8
          CONST drift_delay_counter                 7
          CONST auto_off_counter                    6

          CONST ram_last_offset                     5
          CONST ram_offset_off_flag                 4

          CONST roll_avg_ram_3                      3
          CONST roll_avg_ram_2                      2
          CONST roll_avg_ram_1                      1
          CONST roll_avg_ram_0                      0

;-------------------------------------------------------------------------------------------------------
          CONST lcd_reg_high                        15 + 48    ;
          CONST lcd_reg_mid                         14 + 48    ;
          CONST lcd_reg_low                         13 + 48    ;

;-------------------------------------------------------------------------------------------------------
;         CONST cfg_reg_15                          15  + 48
;         CONST cfg_reg_14                          14  + 48
          CONST cfg_reg_13                          16  + 48

          CONST cfg_reg_12                          12  + 48
          CONST cfg_reg_11                          11 + 48
          CONST cfg_bit_lcd_standby                 12

          CONST cfg_reg_10                          10  + 48
          CONST cfg_reg_09                          9  + 48
          CONST cfg_reg_08                          8  + 48
          CONST cfg_reg_07                          7  + 48
          CONST cfg_reg_06                          6  + 48
          CONST cfg_reg_05                          5  + 48
          CONST cfg_reg_04                          4  + 48
          CONST cfg_reg_03                          3  + 48

          CONST cfg_reg_02                          2  + 48

          CONST cfg_reg_01                          1  + 48
          CONST cfg_bit_tdc_sleepmode               17

          CONST cfg_reg_00                          0  + 48
          CONST cfg_bit_abgl_khz10                  9

;-------------------------------------------------------------------------------------------------------
          CONST state_unit_kg                       1                      ;
;         CONST state_unit_oz                       1                      ;
          CONST state_unit_lb                       0                      ;

;-------------------------------------------------------------------------------------------------------
          CONST target_value_5000                   12500;
          CONST auto_off_delay_zero                 15                     ; Time delay for auto-off at zero AUTO_OFF DELAY
          CONST     auto_off_delay_weight           25

;-------------------------------------------------------------------------------------------------------
; Power On Reset Detection
;-------------------------------------------------------------------------------------------------------

rst_pwr_check:
          ramadr    22
          gotoBitS  r, flg_rstssn, Reset_SSN                  ; Reset button

          gotoBitS  r, flg_rstpwr, Reset_SSN                  ; PWR ON Reset

          gotoBitS  r, flg_wdtalt, Reset_SSN                  ;Reset after watchdog timeout

          ramadr    state_ram                       ; Check to see state of chip when reset occured
          gotoBitS  r, state_bit_low_volt_off, Reset  ; scale was turned off (low bat)
```

```
          goto      main_loop

;-------------------------------------------------------------------------------------------------------------
; Power On or Reset
;-------------------------------------------------------------------------------------------------------------
Reset:
          ramadr    state_ram                     ; Check to see state of chip when reset occured
          bitclr    r, state_bit_low_volt_off     ; Indicate that scale was turned off (low bat)

 Reset_SSN:
          ramadr    turn_off_flag
          clear     r
          ramadr    first_turn_off_flag
          clear     r

          ramadr    cfg_reg_00
          move      r,  0x64820A          ; Konfigreg 0 ->sngl conv time 100 (1,3Hz)
          ramadr    cfg_reg_01
          move      r,  0xAFC803          ; Konfigreg 1
          ramadr    cfg_reg_02
          move      r,  0x008356          ; Config. reg. 2          AVRate 2, cycle time 77, no pptemp
          ramadr    cfg_reg_03
          move      r, 0x000210           ; Config. reg. 3         ; 4 MHz OFF    0 MFake, 1xHB
          ramadr    cfg_reg_11
          move      r,  0x2A18C0          ; Config. reg. 11          Comp. OFF, LCD Stdby

          initTDC
          newcyc

;; Check if voltage is high enough
          ramadr    25                    ; RAM address for Ubatt
          move      z, r
          compare   z, 14                 ; Compare voltage value to limit (appr. 2,35V) (gemessen 2,??V)
          skipNeg   3
          ramadr    state_ram
          bitset    r, state_bit_low_volt_off     ; Indicate that scale turned off because of low bat
          goto      stop_prg                      ; If voltage too low, don't do anything

OK_to_Reset:
          ramadr    ram_last_offset
          clear     r
          ramadr    ram_unstable_counter
          clear     r
          ramadr    state_ram                     ; Check to see state of chip when reset occured
          clear     r                             ; after real power out set status to start
          ramadr    state_ram_tmp_cnt             ; reset counter for rolling average in initial offset
          clear     r
          ramadr    ram_track_offset_cnt
          clear     r
          ramadr    drift_delay_counter           ; reset counter for drift delay
          clear     r
          ramadr    auto_off_counter              ; reset auto-off counter
          clear     r
          ramadr    ram_scale_unit_mode                      ;
          clear     r
          bitset    r, state_unit_kg      ;set display to kilogramm

          ramadr    state_ram
          bitset    r,state_bit_init_offset
          bitset    r, state_bit_first_run

          ramadr    ram_last_display
          clear     r
          ramadr    auto_off_counter
          clear     r                     ; When the display value changes reset auto-off counter

          ramadr    cfg_reg_00                                 ;
          move      r, 0x64828A           ; Config Register 0    ; conv timer 100  disable start curr osc.  cpuspd def
          ramadr    cfg_reg_01
          move      r,  0xADC003          ; Take TDC out of sleep mode
          ramadr    cfg_reg_02
          move      r, 0x0084DE           ; AVrate  2, cytime 77  with pptemp
          ramadr    cfg_reg_03
          move      r, 0x040217           ; Turn on 4MHz osci. (800µs), MFake 1, 4xHB

          ramadr    cfg_reg_11
          move      r,  0x2A08C1          ; Konfigreg 11          -> LCD standby bit 12 clear, comp. to "ON during measurement" ;

main_loop:
;----------------------------- check_voltage ---------------------------------------------------------------------
          ramadr    25                    ; RAM address for Ubatt
          move      z, r
          compare   z, 14                 ; Compare voltage value to limit (appr. 2,35V) (gemessen 2,??V)
          gotoNeg   voltage_OK

          ramadr    state_ram
          bitset    r, state_bit_low_volt_off     ; Indicate that scale turned off because of low bat
                                                          ;;  Voltage is going down!  Turn off all you can !
low_volt_turn_off:
          ramadr    cfg_reg_01
          bitset    r, cfg_bit_tdc_sleepmode      ; Turn off TDC
          ramadr    cfg_reg_03
          move      r, 0x000210                   ; 4 MHz OFF    0 MFake, 1xHB
          ramadr    cfg_reg_11
          bitset    r, cfg_bit_lcd_standby        ; Turn off LCD
          bitclr    r, 0                          ; Clear both bits
          bitclr    r, 1                          ; for comp. control
          goto      stop_prg

voltage_OK:
```

```
;-------------------------------------------------------------------------------------------------------------------------
; Routine to monitor the Offset/creeping
; Zaehler=5
 ; Offset groesser 100g --> Out 2
; Ramadr20 > !Skalenteil -->weitermitteln + unset

;-------------------------------------------------------------------------------------------------------------------------
;--Get Initial Offset Value
            ramadr    state_ram
            gotoBitC  r, state_bit_init_offset,  no_init_offset        ; If no offset value (scale off) skip monitoring

            ramadr    lcd_reg_low
;           clear     r
            move      r, 0x6D7978
            ramadr    lcd_reg_mid
;           clear     r
            move      r, 0x003F71
            ramadr    lcd_reg_high
            clear     r
; Fill Rolling Average Filter
            ramadr    20                      ;HB0 compensated
            move      x, r
;;; Check values to see if step in measured weight (weight was put on scale) - only in scan mode!
            ramadr    state_ram
            gotoBitS  r, state_bit_first_run, no_step

            ramadr    ram_last_value          ; get last value in order to detect jump in weight
            move      z, r
            and       z, 0xFFFFFF
            getflag   z
            otoEQ     no_step                 ; last value is zero -> first time around

            sub       z, x
            abs       z
            compare   z, 12000                ;12000/800 = 1,5kgr
            gotoPos   no_step

            ramadr    state_ram
            bitclr    r, state_bit_init_offset
            bitset    r, state_bit_auto_on_now
            ramadr    ram_last_value                      ; store last value in order to detect jump in weight
            clear     r
            goto      end_init_offset

no_step:
            ramadr    ram_last_value          ; Tara offset is re-used here to store last value in order to detect jump in weight
            move      r, x

            jsub      roll_avg_move_to
;-- Count Loops for Initial Offset
            ramadr    state_ram_tmp_cnt
            incr      r
            compare   r,6                     ; Is it higher than 10?
            gotoPos   end_init_offset

;-- Test Actual Offset
test_offset:
            ramadr    roll_avg_mean_5         ; Previous value from filter
            move      z, r
            ramadr    20                      ;HB0 compensated
            move      x, r
            jsub      roll_avg_move_to        ; Send new value through filter -> filter output in x
            sub       x, z
            abs       x
            compare   x, 500                  ;500/800 = 62,5g
            gotoPos   init_offset_ok

            ramadr    state_ram
            skipBitC  r, state_bit_off, 2
            compare   x, 4000                 ;4000/800 = 500g
            gotoPos   init_offset_ok

            ramadr    ram_unstable_counter
            incr      r
            compare   r, 5
            skipNeg   1
            clrwdt                                        ;clear watchdog
            stop

init_offset_ok:
            ramadr    lcd_reg_mid
            clear     r
            ramadr    lcd_reg_low
            clear     r
            ramadr    lcd_reg_high
            clear     r
;           ramadr    lcd_reg_high
            bitset    r, 5                    ; Display sun to indicate voltage is high enough

;-- Tara / Cal On
            ramadr    roll_avg_mean_5
            move      x,r

; New offset while in OFF mode?
            ramadr    state_ram
            gotoBitC  r, state_bit_init_offset_off, normal_offset    ; NOT in off mode go to normal procedure

;----------  Compare new offset value with previous one. If too far off discard new offset
            ramadr    turn_off_flag                                                                    ;
            gotoBitS  r, 0, store_new_offset ; always take new value when first run
```

```
            ramadr   ram_seg_run_value
            move     y, r
            sub      y, x
            abs      y
            compare  y, 9000              ; 9000/800 = 1,5kg
            skipNeg  2
store_new_offset:
            ramadr   ram_seg_run_value    ; In OFF mode offset is stored here to be recalled when scale turns back ON
            move     r, x

            ramadr   turn_off_flag
            clear    r

            ramadr   state_ram
            clear    r
            bitset   r, state_bit_off

            ramadr   cfg_reg_00                                      ;
            move     r, 0xC8828A          ; conv timer 200 disable start curr osc.  cpuspd def
            ramadr   cfg_reg_02
            move     r, 0x0084D6          ;  AVRate 2  no pptemp
            ramadr   cfg_reg_03
            move     r, 0x040210          ; 4 MHz osc.  200µ    0 MFake  1xHB

            ramadr   state_ram_tmp_cnt    ; reset counter for rolling average in initial offset
            clear    r
            clear    x
            jsub     roll_avg_initialize

            ramadr   ram_last_value
            clear    r

            goto     stop_prg             ; Go back to settings in OFF mode

normal_offset:
;-- Store Actual Offset
            ramadr   ram_initial_offset
            move     r,x
            ramadr   ram_last_value
            clear    r
;-- Clear Rolling Average
            clear    x
            jsub     roll_avg_initialize

;-- Switch to Next state
            ramadr   state_ram_tmp_cnt
            clear    r
            ramadr   state_ram
;           clear    r
            bitset   r,state_bit_gauge
            bitclr   r, state_bit_init_offset

            ramadr   state_ram
            skipBitC r, state_bit_off, 3
            ramadr   cfg_reg_00                                      ;
            move     r, 0xC8828A          ; conv timer 200 disable start curr osc.  cpuspd def
            goto     end_init_offset

            ramadr   cfg_reg_00                                      ;
            move     r, 0x64828A          ; conv timer 100  disable start curr osc.  cpuspd def

            ramadr   state_ram
            bitclr   r, state_bit_first_run

            ramadr   auto_off_counter                  ; reset auto-off counter
            move     r, auto_off_delay_weight          ; Set auto off counter to turn to scan mode immediately after original init offset

end_init_offset:
            newlcd
            clrwdt
            stop

no_init_offset:
            nop

;------------------------------------------------------------------------------------------------------------------------
; Change Scale units gr - oz - dwt - ozt
;------------------------------------------------------------------------------------------------------------------------
            ramadr   22
            gotoBitC r,flg_sdi_r_io1,no_unit_mode_change
            ramadr   ram_scale_unit_mode
            shiftR   r
            skipNE   1                    ; wrap Bit 3..0  around
            bitset   r, state_unit_kg     ;set ram_unit_mode to gramm again
 no_unit_mode_change:
;----------------------------------------------------------------------------------------------------------------------
;The following accumulators are used temporarily to deteremine some parameters
;           x : Loaded with byte that contains the bits for the Unit
;           y : Scalingfactor for Unit
;           z : Positon of comma
;           r  : ram_tmp_unit_digit : LCD Segment to display unit
;
; Please note that they need to contain the following parameters when calling the no2lcd function later:
;           x: actual result to be displayed
;           y: number of digits after the comma
;           z: not used
;----------------------------------------------------------------------------------------------------------------------
            ramadr   ram_scale_unit_mode ; Load the byte with the unit bits into x

            clear    x                    ; clear unit byte
;-- kg
```

```
          skipBitC  r, state_unit_kg, 1      ; use parameters for gr when bit ís set. Otherwise skip 2 cmds
          bitset    x, 2                     ; set digit "kg"
;--lb
          skipBitC  r, state_unit_lb, 1      ; use parameters for oz when bit ís set. Otherwise skip 3 cmds
          bitset    x, 6                     ; set digit "lb"

          ramadr    ram_tmp_unit_digit
          move      r, x                     ; Store byte with the correct bit for displaying the unit set

;-----------------------------------------------------------------------------------------------------
;--- Subtract Initial Offset value from measurement value
;-----------------------------------------------------------------------------------------------------
          ramadr    ram_initial_offset
          move      x, r                     ; Load the initial offset value (unscaled) into x
          ramadr    20                       ; Load the raw measurement value...
          sub       x, r                     ; ...and subtract the initial offset value
;-----------------------------------------------------------------------------------------------------
; Multiplication factor for calibration of displayed value
;-----------------------------------------------------------------------------------------------------
          ramadr    14
          getepr    z
          mult24    x,z                      ; Multiply with cal factor -> result is stored in x
;-----------------------------------------------------------------------------------------------------
; Make sure result is not above upper limit
;-----------------------------------------------------------------------------------------------------
          compare   x, 0xCD000               ; Compare measurement value to upper limit (appr. 10kg)
          gotoPos   no_overload              ; if x is bigger jump to no_overload
          ramadr    lcd_reg_mid              ; Result exceeds upper limit
          move      r, 0x005C38              ; display an error message
          ramadr    lcd_reg_low
          move      r, 0x5C775E
          goto      stop_prg                 ; Skip the rest of the program and go to end
;-----------------------------------------------------------------------------------------------------
; Check supply voltage to make sure measurments are reliable
;-----------------------------------------------------------------------------------------------------
no_overload:
          ramadr    25                       ; RAM address for Ubatt
          move      z, r
          compare   z, 14                    ; Compare voltage value to lower limit (appr. 2,35V) (gemessen 2,??V)
          gotoNeg   no_lowbat                ; if z is bigger jump to no_lowbat
          ramadr    lcd_reg_low              ; Supply voltage is below lower limit
          clear     r
          ramadr    lcd_reg_mid              ; Supply voltage is below lower limit
          clear     r
          ramadr    lcd_reg_high             ; Supply voltage is below lower limit
          compare   z, 21                    ; Compare voltage value to turn off limit (appr. 2,6V) (gemessen 2,62V)
          skipNeg   2
          clear     r
          goto      turn_off_scale           ; Skip the rest of the program and go to end
          ramadr    ram_tmp_unit_digit
          bitset    r, 3
;-----------------------------------------------------------------------------------------------------
; Scale Display to final unit (measurement value is still in x)
;-----------------------------------------------------------------------------------------------------
no_lowbat:
          move      y, 0x7FFFFF              ; Load factor for unit gr into y
          mult24    x,y                      ; Multiply value with unit factor -> result is stored in x-accu

          ramadr    ram_tmp_mlt_hb0
          move      r, x                     ; Store measurement value scaled to unit in "ra_tmp_mlt_hb0"
;-----------------------------------------------------------------------------------------------------
; Feed value into Rolling Average Filter -> Result of filter will be in x
;-----------------------------------------------------------------------------------------------------
          jsub      roll_avg_move_to
;-----------------------------------------------------------------------------------------------------
;--- Step Filter with actual measurement value
;-----------------------------------------------------------------------------------------------------
          ramadr    ram_tmp_mlt_hb0
          move      x, r

          move      y, x
          ramadr    ram_last_display
          sub       y, r
          abs       y                        ; difference to actual value on display
          compare   y,500                    ; step filter width of 3 scale division
          gotoPos   show_on_lcd

;-----------------------------------------------------------------------------------------------------
;--- AUTO ON if value changes enough
;-----------------------------------------------------------------------------------------------------
          ramadr    state_ram
          gotoBitS  r, state_bit_auto_on_now, auto_on         ; If this bit is set go to auto-on no matter what
          gotoBitC  r, state_bit_off, no_auto_on         ; Check for AUTO ON only when scale is OFF

          compare   y, 3000                  ; Value here is in gr -> 3000/1000 = 3kg AUTO-ON THRESHOLD
          gotoPos   no_auto_on

 auto_on:
          ramadr    cfg_reg_00               ;
          move      r, 0x64828A              ; conv timer 100  disable start curr osc.  cpuspd def
          ramadr    cfg_reg_02
          move      r, 0x0084DE              ;  AVRate 2  with pptemp
          ramadr    cfg_reg_03
          move      r, 0x040217              ; 4 MHz osc.  200µ   1 MFake 4xHB
          ramadr    cfg_reg_11
          move      r, 0x2A08C1              ; Comp. ON during measure   LCD ON

          ramadr    drift_delay_counter      ; reset counter for drift delay
          clear     r
          ramadr    auto_off_counter         ; reset auto-off counter
          clear     r
```

```
            ramadr    ram_last_display
            clear     r

            ramadr    ram_last_value
            clear     r

            ramadr    ram_seg_run_value
            move      z, r
            ramadr    ram_initial_offset
            move      r, z

            ramadr    state_ram
;           move      r, 2                 ; Set bit for init reset and clear all else
            move      r, 4                 ; Set bit for measure and clear all else

            clrwdt                         ;clear watchdog
            stop                            ;stop processor
;-----------------------------------------------------------------------------------
;--- Step to Near to Zero
;-----------------------------------------------------------------------------------
no_auto_on:
            ramadr    ram_tmp_mlt_hb0
            skipNeg   1
            clear     r                    ; display zero
;-----------------------------------------------------------------------------------
;--- Initialize rolling average
;-----------------------------------------------------------------------------------
re_init_roll_avg:
            move      x, r
            jsub      roll_avg_initialize

show_on_lcd:
            nop
            nop
;-----------------------------------------------------------------------------------
; Show Measurement Value
;-----------------------------------------------------------------------------------
            ramadr    roll_avg_mean_5
            move      x, r

            move      y, x
            abs       y
            compare   y, 400               ; region around zero, value here is in gr -> 400/1000=0,4kg ZERO HYSTERESIS
            skipNeg   1
            clear     x                    ; display zero

;-----------------------------------------------------------------------------------
;--- Hysteresis
;-----------------------------------------------------------------------------------
            move      y, x
            ramadr    ram_last_display
            sub       y, r
            abs       y
            compare   y, 100               ; value here is in gr -> 100/1000=0,1kg  HYSTERESIS
            skipNeg   1
            move      x, r

;-----------------------------------------------------------------------------------
;--- store displayed value for Hysteresis and Step Filter
;-----------------------------------------------------------------------------------
            ramadr    ram_last_display
            move      z, r                 ; Save last display value in z for auto off
            move      r,x
;-----------------------------------------------------------------------------------
;  Scale to Display
;-----------------------------------------------------------------------------------
            move      y,  10               ;;;; DON't forget to change !
            divmod    x,y
;-----------------------------------------------------------------------------------
;  Scale to Display
;-----------------------------------------------------------------------------------


; Get new offset periodically
            ramadr    state_ram
            gotoBitC  r, state_bit_off, auto_off         ; When in ON mode check for AUTO OFF

            ramadr    state_ram_tmp_cnt
            incr      r
            ramadr    ram_track_offset_cnt
            incr      r
            compare   r, 60                              ; Take new offset appr. every 60sec.
            gotoPos   value_change

            ramadr    first_turn_off_flag
            skipBitC  r, 0, 3
            clear     r
            ramadr    turn_off_flag
            incr      r

            ramadr    state_ram_tmp_cnt
            clear     r
            ramadr    ram_track_offset_cnt
            clear     r
new_init_offset:
            ramadr    auto_off_counter
            clear     r                    ; When the display value changes reset auto-off counter

            ramadr    state_ram
            clear     r
            incr      r                    ; Set bit 0 of state ram to indicate OFF status
```

```
            bitset      r, state_bit_init_offset_off
            bitset      r, state_bit_init_offset

            ramadr      cfg_reg_00                                  ;
            move        r, 0x1E828A             ; conv timer 30  disable start curr osc.  cpuspd def
            ramadr      cfg_reg_02
            move        r, 0x0084DE             ;  AVRate 2  with pptemp
            ramadr      cfg_reg_03
            move        r,  0x040217            ; 4 MHz osc.  200µ   1 MFake 4xHB

            goto        value_change

auto_off:
            ramadr      auto_off_counter
            incr        r

            move        y, x
            abs         y                       ; When weight > 2kg turn off when value the same for some time
            compare     y, 200                  ; Here value is gr / 10 -> 200 / 100 = 2kg
            gotoPos     count                   ; When weight < 1,5kg don't care about changing values just turn off after a fixed time!

            move        y,  10
            divmod      z,y                     ; z is last dispalyed value from above, y is still divisor
            sub         z, x
            abs         z
            compare     z, 150                  ; If change in weight is less than 1,5k turn off like zero (value stable)
            gotoNeg     value_change

            compare     r, auto_off_delay_weight
            gotoPos     no_value_change
count:
            compare     r, auto_off_delay_zero
            gotoPos     no_value_change

turn_off_scale:

            ramadr      ram_initial_offset      ; Save offset for measure to ram_seg_run_value
            move        z, r
            ramadr      ram_seg_run_value
            move        r, z
            ramadr      ram_last_value
            clear       r

            clear       x
            ramadr      state_ram
            clear       r
            incr        r                       ; Set bit 0 of state ram to indicate OFF status
            bitset      r,state_bit_init_offset  ; Set bit for init_offset to get new offset value with new settings
;------------ Take offset immadiately after turning off if close to zero
            ramadr      ram_track_offset_cnt
            move        r, 60
            ramadr      first_turn_off_flag
            incr        r
change_settings:
            ramadr      lcd_reg_low             ;
            clear       r
            ramadr      lcd_reg_mid             ;
            clear       r
            ramadr      lcd_reg_high            ; Clear all digits on display
            clear       r
            bitset      r, 5                    ; Set sun symbol

            ramadr      cfg_reg_00
            move        r, 0x1E828A             ; conv timer 30 disable start curr osc.  cpuspd def
            ramadr      cfg_reg_02
            move        r, 0x0084D6             ;  AVRate 2  no pptemp
            ramadr      cfg_reg_03
            move        r, 0x040210             ; 4 MHz osc.  200µ    0 MFake  1xHB

            ramadr      state_ram_tmp_cnt       ; reset counter for rolling average in initial offset
            clear       r
            ramadr      drift_delay_counter     ; reset counter for drift delay
            clear       r
            ramadr      ram_last_display
            clear       r


value_change:
            ramadr      auto_off_counter
            clear       r                       ; When the display value changes reset auto-off counter
no_value_change:
;-----------------------------------------------------------------------------------------------------------
; Scale Display to final unit (measurement value is still in x)
;-----------------------------------------------------------------------------------------------------------
            ramadr      ram_scale_unit_mode ; Load the byte with the unit bits into x
            gotoBitC    r, state_unit_lb, kilogramm          ; use parameters for oz when bit ís set. Otherwise skip 3 cmds
            move        y, 0x3851EB             ; Load factor for unit lb into y
            mult24      x,y                     ; Multiply value with unit factor -> result is stored in x-accu

            skip        2
kilogramm:
            move        y,  10
            divmod      x,y

;-----------------------------------------------------------------------------------------------------------
; Show Measurement Value on LCD
;-----------------------------------------------------------------------------------------------------------
            clear       y
            incr        y
            no2lcd      x, 1
;--- show minus sign if value is negative ---
```

```
        ramadr   0x3e
        skipPos  1                   ;show MSB of value
        bitset   r, 22               ;equals move r,0x00FF00

;------------------------------------------------------------------------------------------------------------
;- show active unit
;------------------------------------------------------------------------------------------------------------

        ramadr   state_ram
        skipBitC r, state_bit_off, 2
        move     z, 0x20
        skip     2

        ramadr   ram_tmp_unit_digit
        move     z, r

        ramadr   lcd_reg_high
        move     r, z

;------------------------------------------------------------------------------------------------------------
;- show when scale is actually OFF
;------------------------------------------------------------------------------------------------------------
        ramadr   state_ram
        gotoBitC r, state_bit_off, not_scanning
        ramadr   lcd_reg_low
        clear    r
        ramadr   lcd_reg_mid
        clear    r
        ramadr   lcd_reg_high
        move     r, 0x20             ; Show sun symbol
        newlcd

        ramadr   state_ram_tmp_cnt
        move     z, r
        move     y, 1
        move     y, 3
        divmod   z, y
        compare  y, 0
        gotoNE   no_blink
        ramadr   cfg_reg_11
        bitclr   r, 12               ; turn ON LCD
        goto     not_scanning

no_blink:
        ramadr   cfg_reg_11
        bitset   r, 12               ; turn OFF LCD

;------------------------------------------------------------------------------------------------------------
; Thermal Drift Compensation of Loadcell
;------------------------------------------------------------------------------------------------------------
not_scanning:
        ramadr   ram_last_display
        getflag  r
        gotoNE   not_at_initial_offset
        ramadr   drift_delay_counter ; Use a counter to delay the drift compensation
        incr     r
        compare  r, 10               ; wait 10 measurements before applying drift compensation
        gotoPos  not_at_initial_offset ; Not at 10 yet, skip drift compensation
        clear    r                   ; Wait is over clear drfit_delay_counter
        ramadr   ram_initial_offset
        move     x, r
        ramadr   20
        sub      x, r
        sign     x          ; Case if no significant numbers, zu wenig nachkommastellen in ram_initial_offset
        ramadr   ram_initial_offset
        add      r, x
not_at_initial_offset:
        nop

;------------------------------------------------------------------------------------------------------------
;--- stop program ---
stop_prg:
        newlcd
        clrwdt                       ;clear watchdog
        stop                         ;stop processor

;------------------------------------------------------------------------------------------------------------
;--- Initialize rolling average
;------------------------------------------------------------------------------------------------------------
roll_avg_initialize:
        ramadr   roll_avg_ram_0
        move     r, x
        ramadr   roll_avg_ram_1
        move     r, x
        ramadr   roll_avg_ram_2
        move     r, x
        ramadr   roll_avg_ram_3
        move     r, x
        ramadr   roll_avg_mean_5
        move     r, x
        jsubret

;------------------------------------------------------------------------------------------------------------
; move value in x and y accu to  Rolling Average and return averaged value in x
;------------------------------------------------------------------------------------------------------------
roll_avg_move_to:
        move     y, x

        ramadr   roll_avg_ram_0
        add      x, r
        swap     y, r
```

```
        ramadr  roll_avg_ram_1
        add     x, r
        swap    y, r

        ramadr  roll_avg_ram_2
        add     x, r
        swap    y, r

        ramadr  roll_avg_ram_3
        add     x, r
        swap    y, r

        ramadr  roll_avg_mean_5         ;averaged HB0
        move    r, x
        move    y, 5                    ; length of average
        divmod  r, y
        move    x, r
        jsubret
;-------------------------------------------------------------------------------------------------------------------------

; --- EOF ---
```

Include file:

```
;---------------------------------------------------------------
;           File: get_button_edge_into_status.h
;
;           Date:  30-05-2008
;
;           Author: FB
;---------------------------------------------------------------

; Program needs to be included if buttons are used !

; Due to a bug in PS08 final release, bit no.0-7 in status register 22 are not working properly in some cases
; This program recalculates these bits and updates them in status register 22, so that they can used without any restrictions afterwards.
; Please add:  #include "get_button_edges_into_status.h"   after the configuration of PS08 which is in most cases:  #include "config.h"
;
; Program characteristics:
; needs 23 bytes, RAMCELL 47 IS OCCUPIED PERMANENTLY
; ramadresspointer and akku x,y,z are used temporally

        ramadr  22
        move    x , r

        setC
        rotR    x
        shiftR  x , 7
        shiftR  x , 12

        move    y , x
        shiftL  y , 4
        invert  x
        add     x , y

        ramadr   47
        swap    x,r
        invert  x
        and     x,r

        ramadr  22
        and     r, 0xFFFF00
        or      r , x
```