# AN2979
# Application note

## Implementing a simple ADC using the STM8L101xx comparator

## Introduction

This application note gives a simple method for implementing an A/D converter with a minimum amount of external components: one resistor and one capacitor.

The pratical application example described in this document uses the STM8L101xx microcontroller comparator.

# Contents

# 1 Application description

## 1.1 Comparator features

The STM8L101xx microcontroller embeds two zero-crossing comparators sharing the same current bias and the same voltage reference.

This voltage reference can be:

- internal (comparison to ground)
- or external (comparison to a reference pin voltage)

Each comparator is connected to 4 channels which can be used to generate an interrupt, a timer input capture or a timer break. Their polarity can be inverted.

*Note: For external comparison be aware that the STM8L101xx comparator maximum input value is $V_{DD}$ -1.25 V with $V_{DDmax}$ = 3.6 V*

## 1.2 ADC implementation

Each comparator can be used to implement an ADC. This technique is based on a simple principle: the signal to be measured is connected to the non-inverted input and the reference signal is an external signal connected to the inverting input.

In the demonstration software, the comparator used is COMP2 and the channel used as the signal to be measured is the channel 3 (pin PD2).

The reference signal is generated by charging a capacitor through a resistor. While the voltage across the capacitor is being charged, it follows an exponential curve.

This exponential equation has been implemented in the software. The time taken by the capacitor voltage to rise above the voltage value to be converted is used in the charge equation to retrieve the digital conversion value.

**Charge equation:**

$$V_{meas} = V_{DD} \times \left(1 - \exp\left(-\frac{t}{T}\right)\right)$$

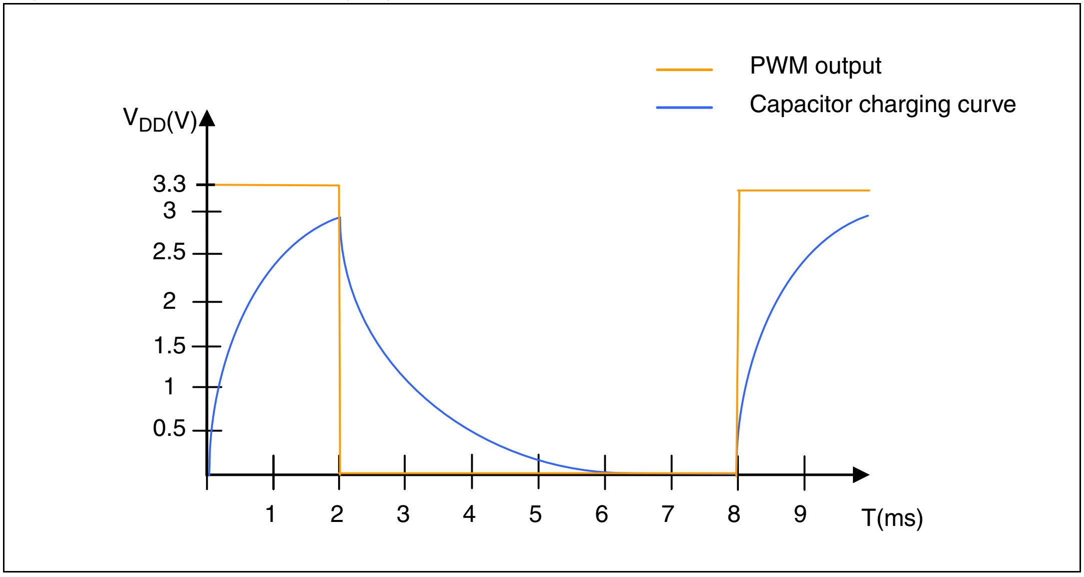
Where:

- **Vmeas** is the value to be mesaured
- **$V_{DD}$** is the input voltage
- **t** is the time measured by timer2 when the comparator detects that the input voltage is above the reference voltage
- **T** is the RC constant (here R=10 kΩ and C=100 nF so T= 1 ms)

The capacitor is charged and discharged using the timer 2 PWM on channel1/ PB0.

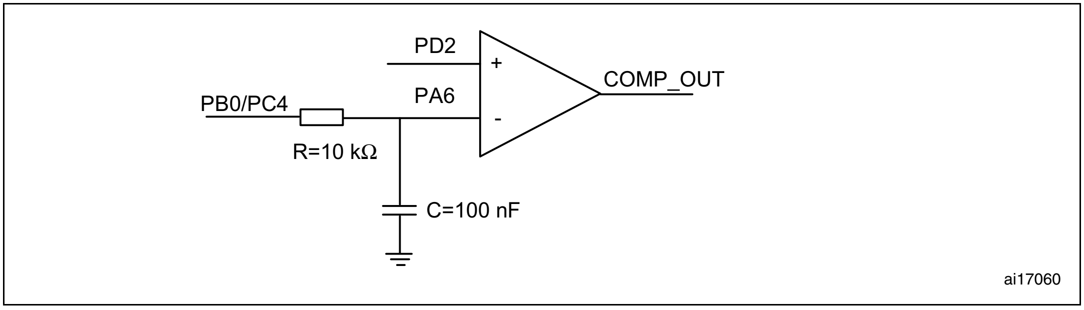
A timer is programmed to generate a 2 ms PWM with a duty cycle of 0.25. This 2-ms period permits an ADC implementation using the full voltage range that the comparator tolerates. The capacitor charging curve is shown in the following figure.

**Figure 1. Capacitor charging curve**



The figure below shows the connection of the comparator to the required external components.

**Figure 2. Comparator connections**



1. Condition: $V_{DD}$=3.3 V, R=10 kΩ, C=100 nF

*Note:* *When using the STM8L101-EVAL evaluation board you need to connect the PB0 pin to the PC4 pin. This way you use the resistor and capacitor already present on the board (no hardware needed). Pin PC4 is configured as input floating to avoid any conflict.*

# 2 Software description

The software provided with this application note describes a way of implementing an A/D converter (using a timer and analog comparator interrupts).

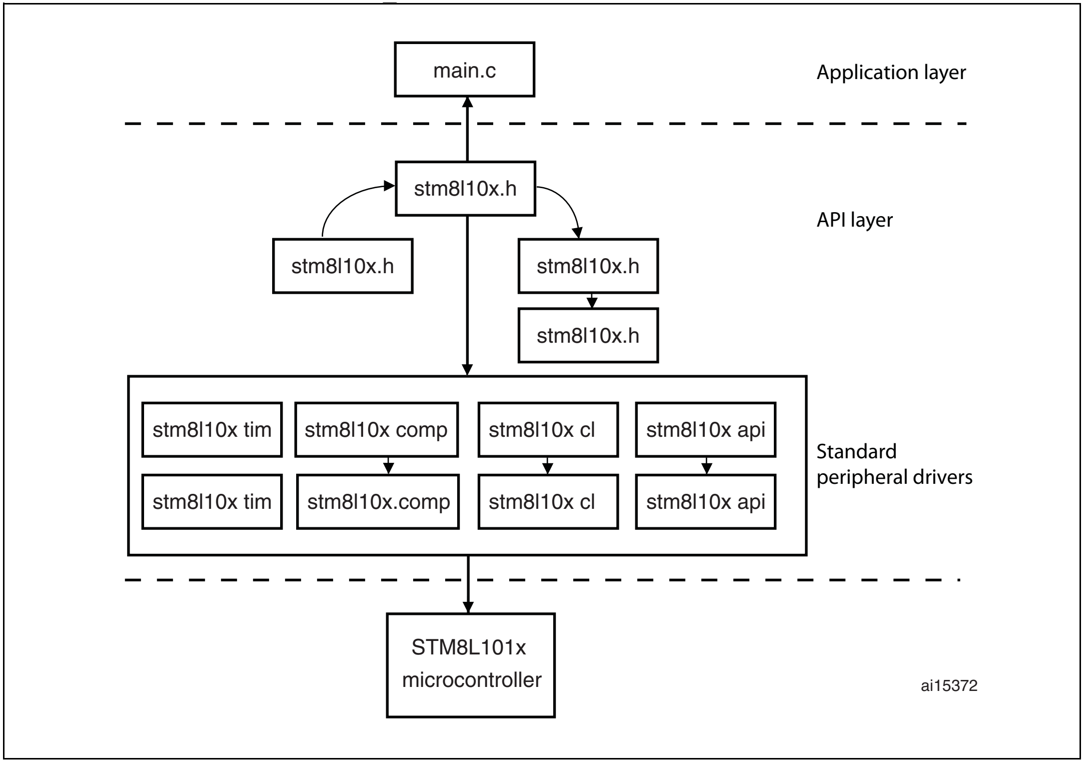
The application uses four peripherals:

- CLK: the clock enables and provides the correct clock frequency for the peripherals.
- COMP: the comparator detects when the input voltage is above the reference voltage
- TIM2: Timer 2 provides the 2 ms PWM with a duty cycle of 0.25 generating the capacitor charge and discharge. Timer 2 Capture/compare interrupt routine handles the conversion.
- GPIO: the general purpose I/O handles the I/O used.
- SPI: used to communicate with the LCD.

A generic file param.h contains the parameter values that can be modified in order to re-use the application and adapt it easily to other conditions.

At the start, the LCD displays “STM8L ADC using COMP”. Then it continously displays the measured values.

**Figure 3. Application architecture & description**

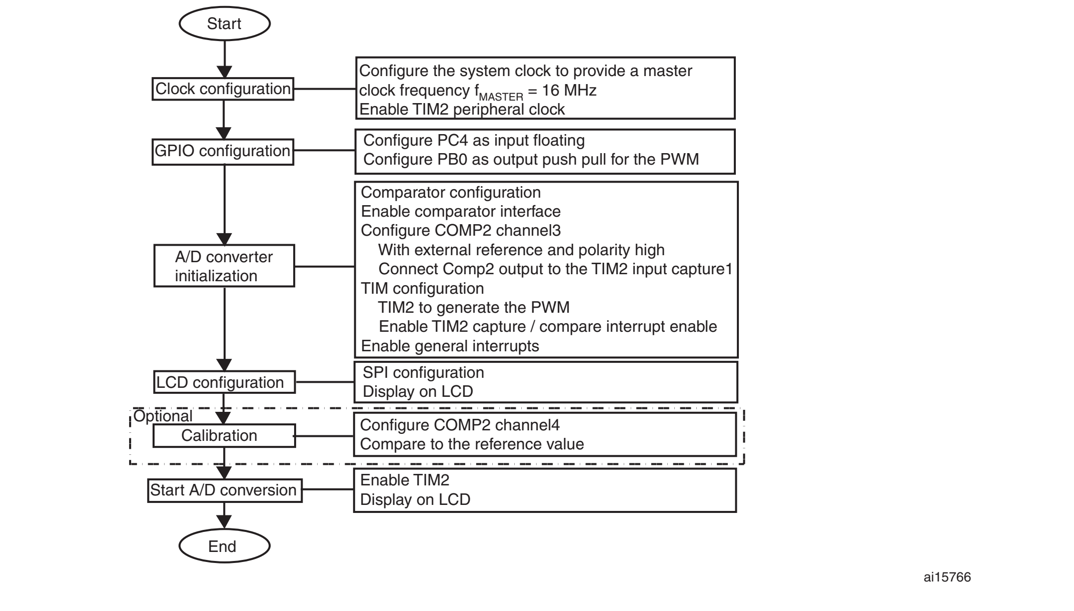


*Note: The software can be compiled with Cosmic and Raisonance compilers. It contains projects for STVD and Raisonance IDE.*

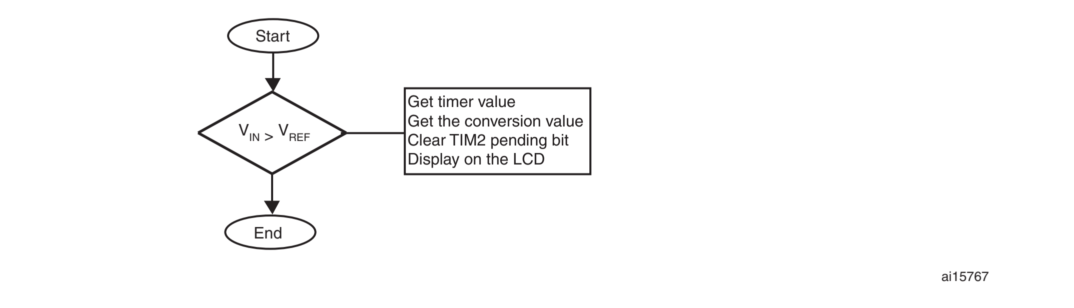
A calibration of some parameters can be optionally set if “#define Calibration” is uncommented in the param.h file.

## 2.1 Application flowcharts

### Figure 4. Main loop flowchart

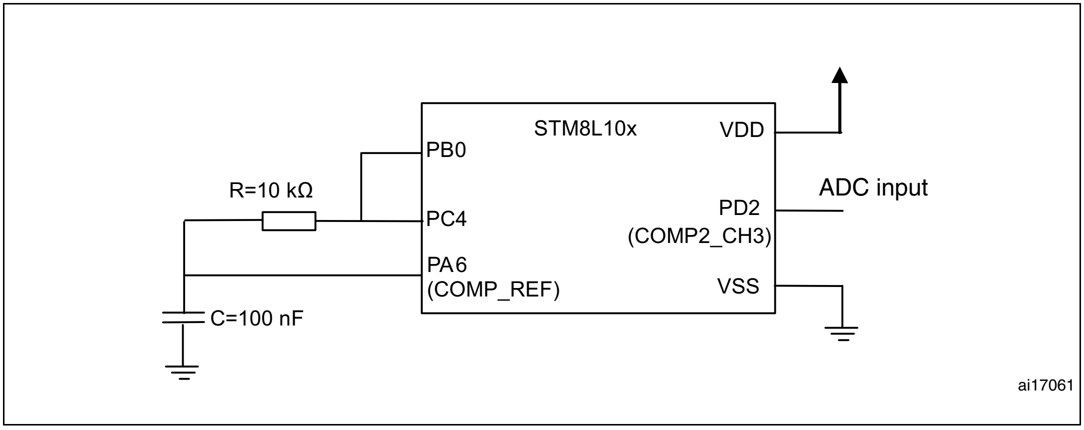

Start
Clock configuration
Configure the system clock to provide a master clock frequency $f_{MASTER}$ = 16 MHz
Enable TIM2 peripheral clock
GPIO configuration
Configure PC4 as input floating
Configure PB0 as output push pull for the PWM
A/D converter initialization
Comparator configuration
Enable comparator interface
Configure COMP2 channel3
With external reference and polarity high
Connect Comp2 output to the TIM2 input capture1
TIM configuration
TIM2 to generate the PWM
Enable TIM2 capture / compare interrupt enable
Enable general interrupts
LCD configuration
SPI configuration
Display on LCD
Optional
Calibration
Configure COMP2 channel4
Compare to the reference value
Start A/D conversion
Enable TIM2
Display on LCD
End
ai15766

### Figure 5. Get conversion value flowchart



# 3 Hardware description

**Figure 6. Circuit diagram**



**STM8L101 evaluation board settings:**

- Remove jumper JP2 to connect an external input voltage instead of the potentiometer input. With the potentiometer RV1, the value is limited to 0.8 V and the signal is noisy (due to the LCD). So the display on the LCD screen would not be stable.
- Connect the external ADC voltage input directly to PD2.
- Ground together the evaluation board and the external ADC signal.
- An accurate 3.3 V MCU voltage can be tuned on the evaluation board using RV4.

# 4 Measurements and calibration

## 4.1 Typical measurements

The following values are given for information only.

Table 1. Measured value after conversion and margin error for $V_{DD}$=3.3 V

| Value to be measured (V) | Measured value after conversion (V) | Error (%) |
|---|---|---|
| 0.25 | 0.204 | 18.4% |
| 0.5 | 0.419 | 16.2% |
| 0.8 | 0.704 | 12% |
| 1 | 0.903 | 9.7% |
| 1.25 | 1.14 | 8.8% |
| 1.5 | 1.39 | 7.33% |
| 1.75 | 1.628 | 6.97% |

## 4.2 Precision of the measured value

When using the RC charge equation, the precision of the measured value depends on the accuracy of the capacitor C and the resistor R.

**Accuracy example:**

if C=100 nF with 10% accuracy: $C_{acc1}$ =90 nF or $C_{acc2}$ =110 nF

if R=10 kΩ with 2% accuracy: $R_{acc1}$=9.8 kΩ or $R_{acc2}$=10.2 kΩ

**Calculation of the RC constant**:

$$T = R \times C = (10 \times 10^{3}) \times (100 \times 10^{-9}) = 1\ \text{ms}$$

$$T_{ACC1} = R_{ACC1} \times C_{ACC1} = (9.8 \times 10^{3}) \times (90 \times 10^{-9}) = 882\ \mu\text{s}$$

$$T_{ACC2} = R_{ACC2} \times C_{ACC2} = (10.2 \times 10^{3}) \times (110 \times 10^{-9}) = 1.122\ \text{ms}$$

**Charge equations:**

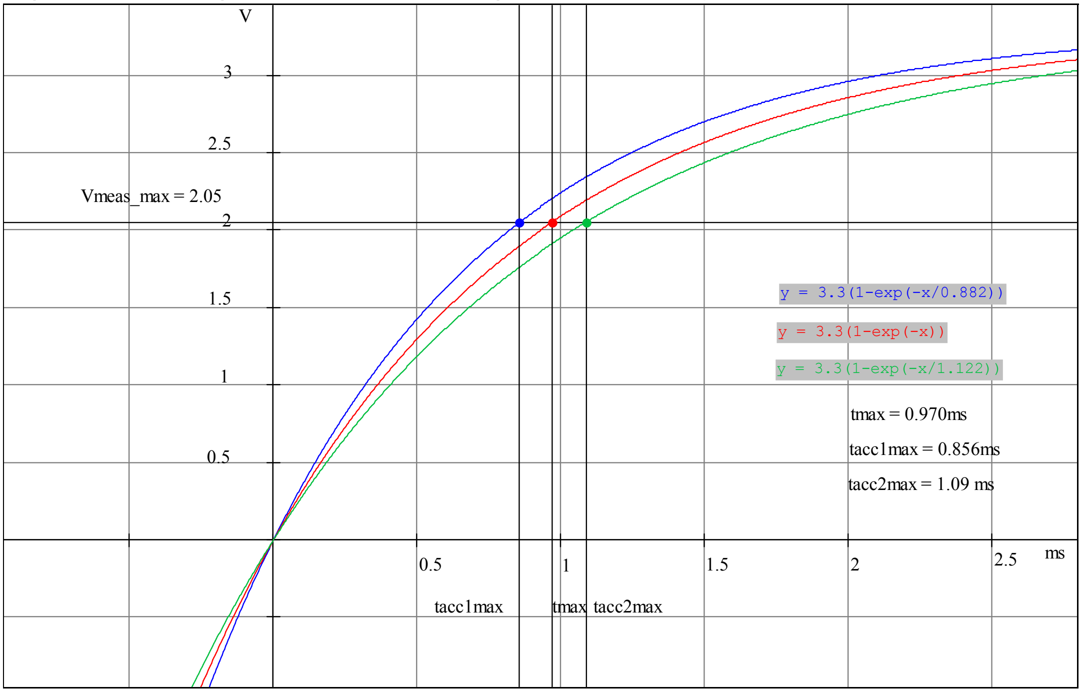
$$V_{meas} = V_{DD} \times \left(1 - \exp\left(-\frac{t}{T}\right)\right)$$

$$V_{measacc1} = V_{DD} \times \left(1 - \exp\left(-\frac{t}{T_{acc1}}\right)\right)$$

$$V_{measacc2} = V_{DD} \times \left(1 - \exp\left(-\frac{t}{T_{acc2}}\right)\right)$$

The following figure shows the impact of the capacitor and resistor accuracy.

**Figure 7. Charge equation depending on RC accuracy**



1. In blue: $V_{measacc1}$ ---- In red: $V_{meas}$ ---- In green: $V_{measacc2}$
2. Condition: $V_{DD}$=3.3 V

*Note: The results are not only impacted by the marging error due to hardware components but also by the accuracy of the MCU clock. In this example, the device works at a HSI frequency of 16 MHz at 25 °C, that is, with a clock accuracy of 1% (refer to the datasheet).*
*The temperature variation also impacts the accuracy of the measured value as the microcontroller, resistors and capacitors are temperature dependent.*

## 4.3 How to get a better accuracy

### 4.3.1 Hardware solution

The first solution is to choose components with a high accuracy value (this may imply a higher cost).

### 4.3.2 Software solution

The second solution is to calibrate the RC constant in the software using $V_{DD}$ as a reference.

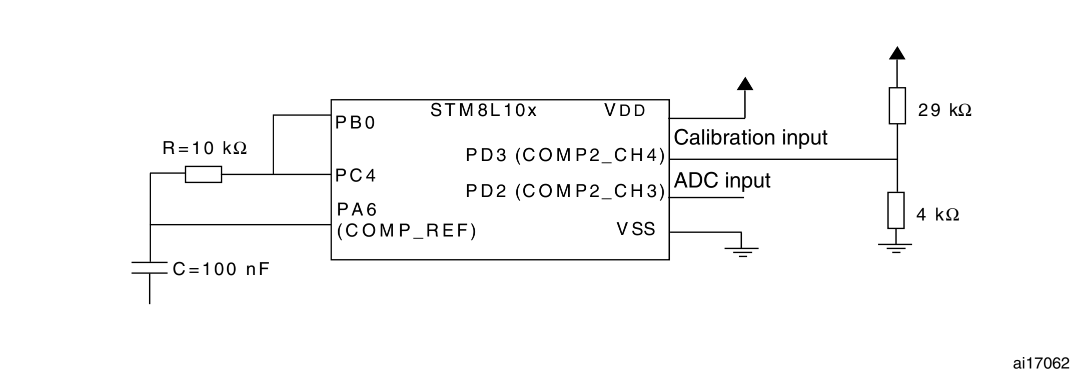
#### Implementation description

In order to perform the calibration, you need to uncomment the “#define calibration” in the “param.h” file.

The calibration is then performed by firware and the RC constant is updated. This constant is used in the charge equation.

On the STM8L101-EVAL evaluation board, the COMP2 channel 4 (PD3) is connected to $V_{DD}$ via a resistor bridge.

Figure 8. Evaluation board calibration schematic



As $V_{DD}$=3.3 V, PD3 is a fixed value equal to 0.4 V

In the file “param.h” you need to define the “expected_value” equal to 0.4 ( this value needs to be updated if a different resistor bridge is used).

The specific function “Calibration()” configures the COMP2 Channel 4 and compares the value measured on this channel with the expected value. This function then updates the RC constant that will be used afterwards.

When the calibration is performed, the COMP2 configuration changes to use the Channel 3 (PD2).

The following table lists a few measurement examples with a calibration performed at $V_{DD}$=3.3 V, R=10 kΩ and C=100 nF.

Table 2. Accuracy measurement when the calibration is performed

| Value to be measured (V) | Measured value after conversion (V) | Error (%) |
|---|---|---|
| 0.25 | 0.233 | 6.8% |
| 0.5 | 0.475 | 5% |
| 0.8 | 0.792 | 1.25% |
| 1 | 0.99 | 1% |
| 1.2 | 1.211 | 0.92% |
| 1.5 | 1.512 | 0.8% |
| 1.75 | 1.784 | 1.9% |

# 5 Revision history

Table 3. Document revision history

| Date | Revision | Changes |
|---|---|---|
| 12-Feb-2010 | 1 | Intitial release. |

STMicroelectronics group of companies

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

**www.st.com**