Specification document of TC1047, TC1047A

Component manufacturer	Microchip Technology		
Model number	TC1047, TC1047A		
Datasheets	21498D.book (microchip.com)		
Specification Ver	01.00.00	Oct 13,2022	New release
	01.00.01	Oct 18,2022	Corrected license content
Documentation provided	Rui Long Lab Inc. https://rui-long-lab.com/		

1. Component datasheet

Temperature accuracy	±2.0° C (Max at +25° C)
Temperature range	-40 to +125° C
Range of power supply voltage (Vdd)	2.7 to 4.4 [V] TC1047
	2.5 to 5.5 [V] TC1047A
Output voltage (Vout)	Linear 0.01 [mV/° C] Typ.
	0 [° C] 0.5 [V] Typ.
Calculation	Vout = 0.5V + (0.01 V/° C × Ta)
	Ta = (Vout – 0.5V) / (0.01 V/° C)

Applications IoT etc

- Cellular Phones
- Power Supply Thermal Shutdown
- Temperature-Controlled Fans
- Temperature Measurement/Instrumentation
- Temperature Regulators
- Consumer Electronics
- Portable Battery-Powered Equipment

2. Component Software IF specification

The software interface specifications based on the TC1047, TC1047A component specifications are as follows.

The voltage value-to-physical value conversion equation is a linear conversion equation as shown in the equation below.

ADC value to voltage value conversion formula

$$vi = (ai \times iADC\_vdd) / 2^{iADC\_bit} \quad [V]$$

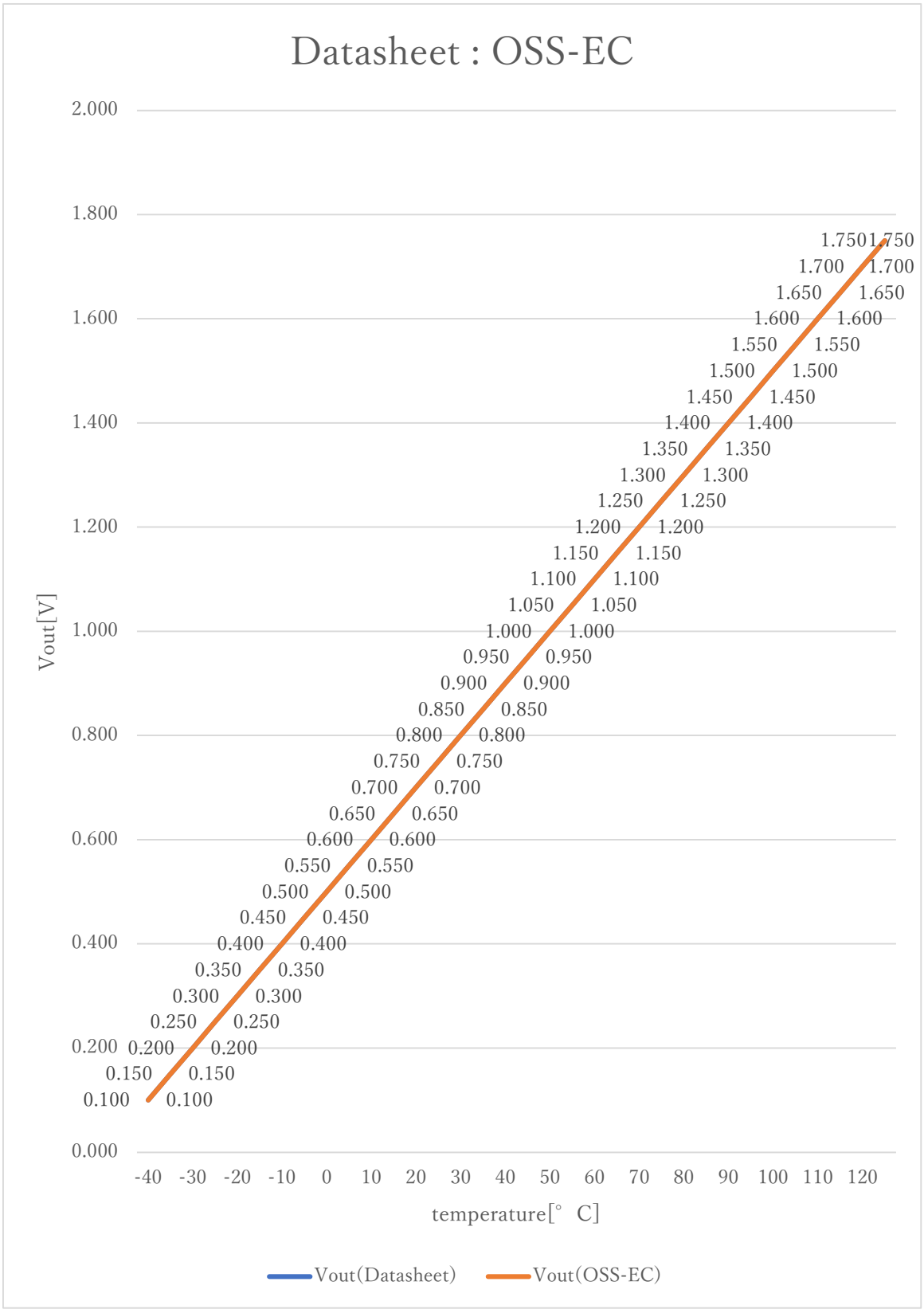
Voltage value to physical value conversion formula

$$y = (vi - iTC1047\_xoff) / iTC1047\_gain + iTC1047\_yoff \quad [℃]$$

$$iTC1047\_min \leqq y \leqq iTC1047\_max$$

ai	A/D conversion value
vi	Sensor output voltage value [V]
iADC_vdd	Sensor supply voltage value [V]
iADC_bit	A/D conversion bit length
y	Temperature value [℃]

```
#define iTC1047_xoff            0.5F            // X offset [V]
#define iTC1047_yoff            0.0F            // Y offset [℃]
#define iTC1047_gain            0.01F           // Gain [V/℃]
#define iTC1047_max             125.0F          // Temperature Max [℃]
#define iTC1047_min             -40.0F          // Temperature Min [℃]
```


Datasheet : OSS-EC
Vout[V]
temperature[° C]
Vout(Datasheet)
Vout(OSS-EC)

3. File Structure and Definitions

TC1047.h

```
#include "user_define.h"

// Components number
#define iTC1047             117U                    // Microchip Technology TC1047, TC1047A

// TC1047 System Parts definitions
#define iTC1047_xoff        0.5F                    // X offset [V]
#define iTC1047_yoff        0.0F                    // Y offset [℃]
#define iTC1047_gain        0.01F                   // Gain [V/℃]
#define iTC1047_max         125.0F                  // Temperature Max [℃]
#define iTC1047_min         -40.0F                  // Temperature Min [℃]

extern const tbl_adc_t tbl_TC1047;
```

TC1047.cpp

```
#include        "TC1047.h"
#if     iTC1047_ma == iSMA                              // Simple moving average filter
static float32 TC1047_sma_buf[iTC1047_SMA_num];
static const sma_f32_t TC1047_Phy_SMA =
{
        iInitial ,                                      // Initial state
        iTC1047_SMA_num ,                           // Simple moving average number & buf size
        0U ,                                            // buffer position
        0.0F ,                                          // sum
        &TC1047_sma_buf[0]                              // buffer
};
#elif   iTC1047_ma == iEMA                              // Exponential moving average filter
static const ema_f32_t TC1047_Phy_EMA =
{
        iInitial ,                                      // Initial state
        0.0F ,                                          // Xn-1
        iTC1047_EMA_K                                   // Exponential smoothing factor
};
#elif   iTC1047_ma == iWMA                              // Weighted moving average filter
static float32 TC1047_wma_buf[iTC1047_WMA_num];
static const wma_f32_t TC1047_Phy_WMA =
{
        iInitial ,                                      // Initial state
        iTC1047_WMA_num ,                       // Weighted moving average number & buf size
        0U ,                                            // buffer poition
        iTC1047_WMA_num * (iTC1047_WMA_num + 1)/2 ,     // kn sum
        &TC1047_wma_buf[0]                              // Xn buffer
};
#else                                                   // Non-moving average filter
#endif

#define iDummy_adr      0xffffffff                      // Dummy address
```

```
const tbl_adc_t tbl_TC1047 =
{
        iTC1047                 ,
        iTC1047_pin             ,
        iTC1047_xoff            ,
        iTC1047_yoff            ,
        iTC1047_gain            ,
        iTC1047_max             ,
        iTC1047_min             ,
        iTC1047_ma              ,

#if     iTC1047_ma == iSMA                      // Simple moving average filter
        &TC1047_Phy_SMA         ,
        (ema_f32_t*)iDummy_adr ,
        (wma_f32_t*)iDummy_adr
#elif   iTC1047_ma == iEMA                      // Exponential moving average filter
        (sma_f32_t*)iDummy_adr ,
        &TC1047_Phy_EMA         ,
        (wma_f32_t*)iDummy_adr
#elif   iTC1047_ma == iWMA                      // Weighted moving average filter
        (sma_f32_t*)iDummy_adr ,
        (ema_f32_t*)iDummy_adr ,
        &TC1047_Phy_WMA
#else                                           // Non-moving average filter
        (sma_f32_t*)iDummy_adr ,
        (ema_f32_t*)iDummy_adr ,
        (wma_f32_t*)iDummy_adr
#endif

};
```