



# Time-to-Digital-Converter

## Application Note

### **TDC-GP22**

**Laser Range Finder with the  
TDC-GP22 Time-to-Digital-Converter**

28<sup>th</sup> January 2014

Document-No: ANO34\_en.pdf V1.3

**Published by acam-messelectronic gmbh**

©acam-messelectronic gmbh 2013

### **Disclaimer / Notes**

“Preliminary” product information describes a product which is not in full production so that full information about the product is not available yet. Therefore, acam-messelectronic gmbh (“acam”) reserves the right to modify this product without notice. The information provided by this data sheet is believed to be accurate and reliable. However, no responsibility is assumed by acam for its use, nor for any infringements of patents or other rights of third parties that may result from its use. The information is subject to change without notice and is provided “as is” without warranty of any kind (expressed or implied). All other brand and product names in this document are trademarks or service marks of their respective owners.

### **Support / Contact**

For a complete listing of Direct Sales, Distributor and Sales Representative contacts, visit the acam web site at:

<http://www.acam.de/sales/distributors/>

For technical support you can contact the acam support team in the headquarters in Germany or the Distributor in your country. The contact details of acam in Germany are:

support@acam.de

or by phone

+49-7244-74190.

## Content

1	Introduction .....	2
1.1	Time-of-Flight.....	3
2	Measurement Range 0 to 300m.....	5
2.1	Single Measurement with Separate Calibration .....	6
2.2	Single Measurement with Auto-Calibration.....	9
3	Measurement Range from 75 m to kilometers .....	11
4	Wide-Range Application.....	14
5	Example .....	17
5.1	Detailed Flow Chart .....	17
5.2	Example Code .....	18
6	Miscellaneous .....	23
6.1	References .....	23
6.2	Literature Guide.....	24
6.3	Document History .....	24

## 1 Introduction

There are several different approaches to build a laser rangefinder. They are similar in that they all use some form of light pulse that is sent out and then somehow received. Once the return pulse has been received, some simple mathematical or geometrical formulas are used to calculate the distance traveled. It's the method of receiving and the fundamentals associated with calculating the distance where these different methods differ.

Some methods are for example:

- Triangulation
- Modulated Continuous Wave
- Time-of-Flight
- ...

Each of these different methods has their own set of advantages and disadvantages.

	Measuring Range	Accuracy	Comment
Triangulation	< 10 m	several $\mu\text{m}$	depending on the surface, inexpensive, robust
Modulated Continuous Wave	< 200 m	approx.. 10 cm	low production costs, slow measurement
Time-of-Flight	several km	several mm	short reaction time, expensive, no aperture

Note for eye-safety



The longer the distance the weaker is the reflected signal. The aim is to increase the laser power. But due to eye safety regulations the total laser power is limited. Therefore, it is mandatory to use pulsed lasers for long distance applications. The shorter the laser pulse width the higher the peak amplitude can be.

This application note describes the implementation of a TDC in a ToF laser rangefinder that is based on acam's TDC-GP22 Time-to-Digital Converter. It describes only the use of the TDC, not the optical part and analog electronics.

## 1.1 Time-of-Flight

Time-of-Flight – this is the time taken for a light pulse to travel to the target and back. With the speed of light known, and an accurate measurement of the time taken, the distance can be calculated. Many pulses are fired sequentially and the average response time is most commonly used. This technique requires very accurate sub-nanosecond timing circuitry.

Measuring distances with lasers or laser scanners is well established – you will find this technique in geodesic systems, security systems, production control systems and even with golf players. Different methods are available based in the distance to be measured. Small distances are measured by triangulation. Using this method the achieved resolution will be in the range of micrometers, but the maximum range is limited to only a few meters. Up to distances of 100 m people often use the phase shift measurement technique. The laser light is modulated and the phase shift between outgoing and incoming light gives the distance. To reach a resolution in the millimeter range very high sampling rates are necessary. Only low measurement rates with high current consumption are possible.

With Time-to-Digital-Converters (TDC's) you choose a direct way to digitize the Time-of-Flight. It is possible to measure the time-of-flight of light directly. The principle is easy - the details are tricky. It is well known that the velocity of light is very high.

$$c = 299\,792\,458 \text{ m/s}$$

$$c := \textit{velocity of light}$$

Because of this, one has to deal with very short times. In only one microsecond the light passes 300 meters! High resolution therefore means highest precision in time measurement. Usually the ray of light will be reflected off an object or mirror. Therefore the light passes the distance twice. So we will have:

$$\textit{distance} = \frac{c}{2} * \textit{ToF}$$

Typical data:

ToF	Distance
66.67 $\mu$ s	10 km
2.4 $\mu$ s	360 m
6.6 ns	1 m
66 ps	10 mm
10 ps	1.5 mm
6.6 ps	1 mm

This is the range well suited for use of the TDC-GP22. This single-chip TDC has a single-shot resolution 90 ps which is equivalent to 13.5 mm distance. The resolution can be further improved by averaging down to <10 ps or <1.5 mm.

The TDC-GP22 is a very universal integrated circuit and can be used in different modes. Which mode should be chosen? In the following we look at some modes in detail:

## 2 Measurement Range 0 to 300m

TDC-GP22 – Measurement Mode 1

### General Description

- Measurement range 3.5 ns to 2.0  $\mu$ s (0 to 2.0  $\mu$ s between stop channels)
- Typ. 90 ps resolution rms (13.5 mm)
- Further resolution improvement by averaging
- Up to 500,000 measurements per second
- 20 ns pulse pair resolution
- 4-fold multihit capability for each stop channel
- Selectable rising/falling edge sensitivity for each channel
- Enable pins for windowing functionality
- Typical application: Handheld rangefinders up to 300 m, speed guns

### Block Diagram

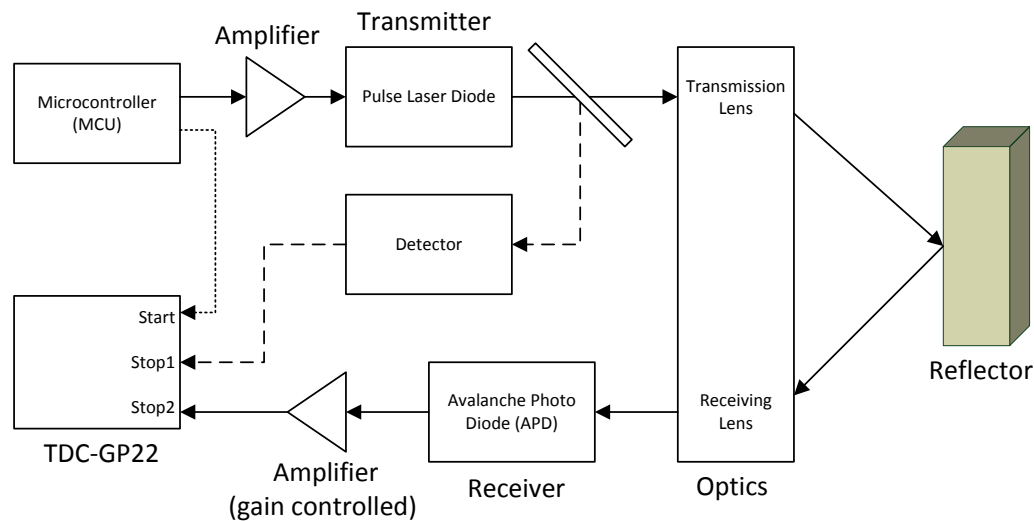


Figure 2.1 Block Diagram – Measurement Mode 1

Temperature effects on the transmitting and receiving paths are eliminated by measuring the twin interval between a reference beam and the reflected beam. So the TDC gets a fake start from the microcontroller. The optical reference is given to Stop1, the reflected beam to Stop2.

## Advantages:

- Measurement down to zero ns
- Averaging gives real improvement with activated noise unit (EN\_STARTNOISE = "1").  
With the artificial start noise enabled, the TDC has the necessary statistics to improve resolution by averaging.

## 2.1 Single Measurement with Separate Calibration

Separate calibration by sending opcode Start\_Cal\_TDC via the SPI interface.

The time measurement raw data is internally stored. The number of hits can be seen from the status register, bits 3 to 8. In case calibration is active the TDC now measures one and two periods of the internal reference clock ( $T_{ref} * 1, 2 \text{ or } 4$ ). The calibration raw data Cal1 and Cal2 are also internally stored.

### Timing – Single Measurement without Calibration

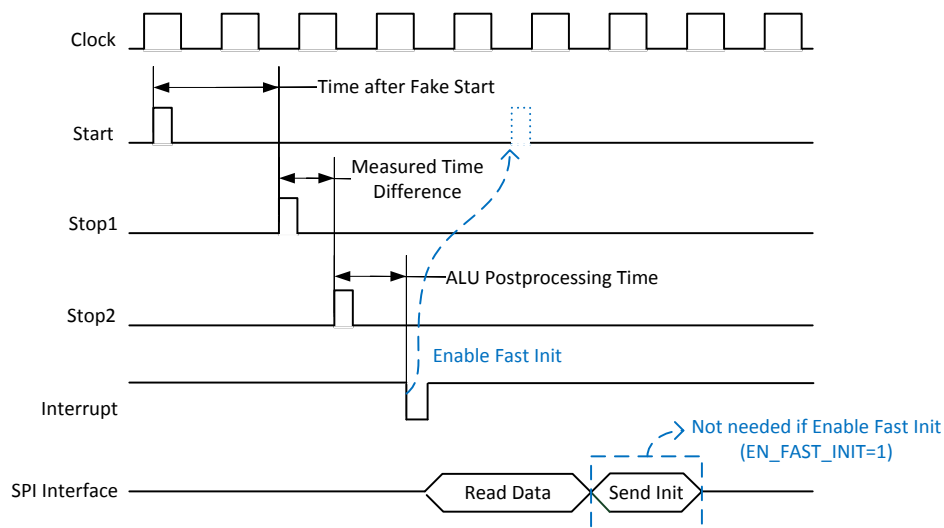


Figure 2.2 Timing – Single Measurement without Calibration

TDC-GP22 offers the possibility to configure a fast initialization. Therefore, EN\_FAST\_INIT = 1.

With this setting the TDC is automatically re-initialized as soon as the results have been calculated. It is no longer necessary to send an INIT opcode. This is helpful for high data rate applications.



## Timing – Separate Calibration

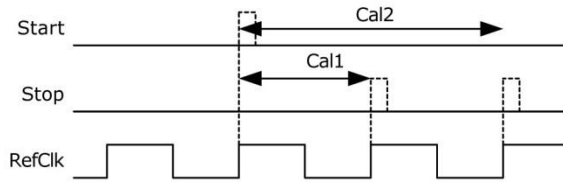


Figure 2.3 Timing – Separate Calibration

### Settings

Set register 0, bit 11, MESSB2 = 0

Select measurement mode 1

Set register 5, bit 28, EN\_STARTNOISE = 1

Switch on additional noise unit for start channel. It is dedicated to applications where the TDC gets a dummy start and measures the time difference between STOP1 and STOP2.

Set register 1, bit 15, EN\_FAST\_INIT = 1

The TDC is already prepared for the next measurement while the data is read out. This mode is for highest speed applications only. It is most reasonable for un-calibrated measurements with only one stop.

Configure Chip 1 (Measurement Mode 1):

Register	Value	Typical example configuration
Register 0	'h00241000	DIV_CLKHS = 2, the 4 MHz ceramic oscillator is internally divided by 4, internal period = 1 $\mu$ s, maximum measurement time $2 \times T_{ref} = 2 \mu$ s START_CLKHS = 1, Oscillator continuously on CALIBRATE = 0, mandatory in measurement mode 2 to be on NO_CAL_AUTO = 1, mandatory in measurement mode 2 to have auto-calibration MESSB2 = 0, measurement mode 1 for measuring < 2 $\mu$ s. NEG_STOP/NEG_START = 0, all set to rising edges
Register 1	'h19C90000	HIT2 = 1, HIT1 = 9: calculate 1.Stop CH2 – 1.Stop CH1 in measurement mode 1 EN_FAST_INIT = 1, enables fast init operation HITIN2 = 1, one expected hit on CH2 HITIN1 = 1, one expected hit on CH1
Register 2	'hA0000000	EN_INT = b101, interrupt given by Timeout or ALU ready. (see also register 6) RFEDGE1 = RFEDGE2 = 0, use only rising edges

Register 3	'h00000000	
Register 4	'h20000000	
Register 5	'h10000000	CON_FIRE = b000, output configuration for pulse generator EN_STARTNOISE = 1, switch on
Register 6	'h00000000	

For further setting of the TDC-GP22, please use our data sheet.

### 2.1.1 Averaging and Separate Calibration

Better resolution by averaging, so higher sampling rate and therefore only occasional calibration.

#### Principle FlowChart

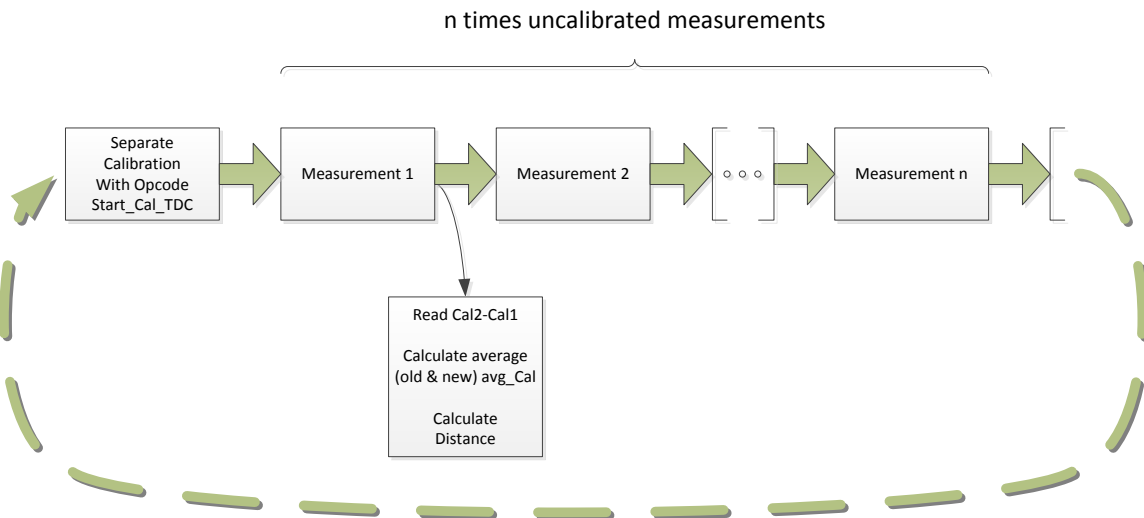


Figure 2.4 Principle Averaging and Separate Calibration FlowChart

$$Distance = average\ ToF\ Result * \frac{T_{Ref}}{average\ (Cal2 - Cal1)\ value} * \frac{velocity\ of\ light}{2}$$

## 2.2 Single Measurement with Auto-Calibration

In most applications automatic calibration is the preferred setting.

In this mode the user does not need take care of calibration. Everything is done automatically by the TDC itself. But the measurement rate is limited in this mode.

### Timing

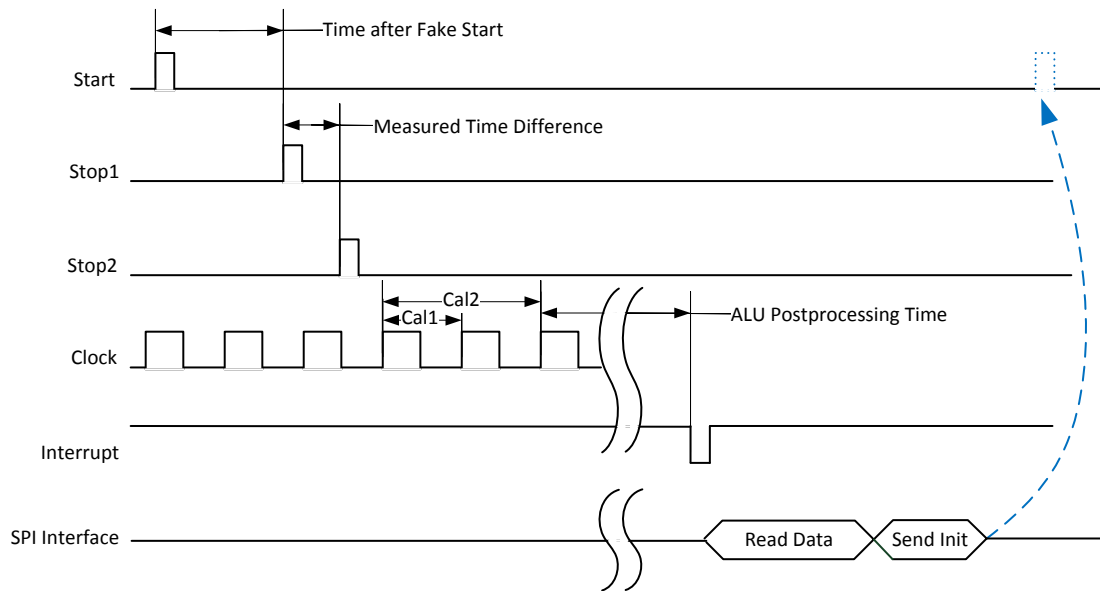


Figure 2.5 Timing – Single Measurement with Auto-Calibration

### Settings

Set register 0, bit 11, MESSB2 = 0

Select measurement mode 1

Set register 0, bit 12, NO\_CAL\_AUTO = 0

Enable Automatic Calibration.

Set register 0, bit 13, CALIBRATE = 1

Enable calibration calculation in the ALU.

Set register 5, bit 28, EN\_STARTNOISE = 1

Switch on additional noise unit for start channel. It is dedicated to applications where the TDC gets a dummy start and measures the time difference between STOP1 and STOP2.

Configure Chip 1 (Measurement Mode 1):

Register	Value	Typical example configuration
Register 0	'h00242000	DIV_CLKHS = 2, the 4 MHz ceramic oscillator is internally divided by 4, internal period = 1 $\mu$ s, maximum measurement time $2 \times T_{ref} = 2 \mu$ s START_CLKHS = 1, Oscillator continuously on CALIBRATE = 1, mandatory in measurement mode 2 to be on NO_CAL_AUTO = 0, mandatory in measurement mode 2 to have auto-calibration MESSB2 = 0, measurement mode 1 for measuring < 2 $\mu$ s. NEG_STOP/NEG_START = 0, all set to rising edges
Register 1	'h19490000	HIT2 = 1, HIT1 = 9: calculate 1.Stop CH2 – 1.Stop CH1 in measurement mode 1 EN_FAST_INIT = 0, off HITIN2 = 1, one expected hit on CH2 HITIN1 = 1, one expected hit on CH1
Register 2	'hE0000000	EN_INT = b111, interrupt given by Timeout, End Hits or ALU ready. (see also register 6) RFEDGE1 = RFEDGE2 = 0, use only rising edges
Register 3	'h00000000	
Register 4	'h20000000	
Register 5	'h10000000	CON_FIRE = b000, output configuration for pulse generator EN_STARTNOISE = 1, switch on
Register 6	'h00000000	

For further setting of the TDC-GP22, please use our data sheet.

### 3 Measurement Range from 75 m to kilometers

#### TDC-GP22 – Measurement Mode 2

Measurement Mode 2 offers long range measurement, up to many kilometers. But it is limited at the lower end. Measurement Mode 2 needs minimum  $2 \cdot T_{ref}$  between Start and Stop1.

#### General Description

- 1 stop channel referring to one start channel
- Typical 22 ps / 45 ps / 90 ps resolution
- Measurement range  $2 \times T_{ref}$  to 4 ms @ 4MHz
- $2 \times T_{ref}$  pulse pair resolution
- 3-fold multihit capability, fully-automated calculation
- Selectable rising/falling edge sensitivity
- Integrated programmable windowing for each single stop with 10 ns precision
- Typical application: Long distance range finder for, hunting, golf, geodesic instruments.

#### Block Diagram

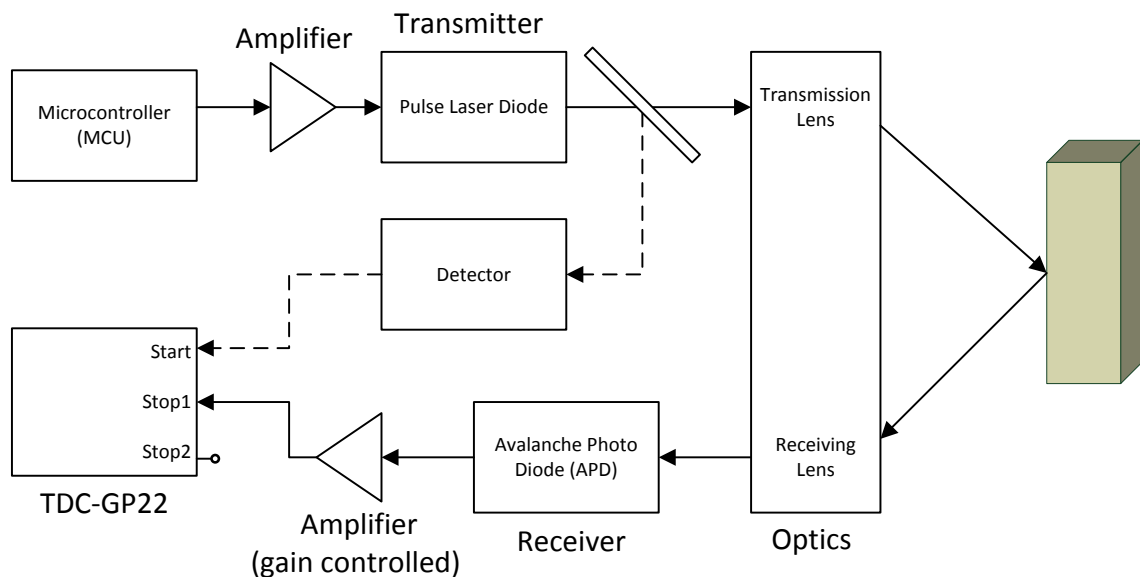


Figure 3.1 Block Diagram – Measurement Mode 2

## Timing

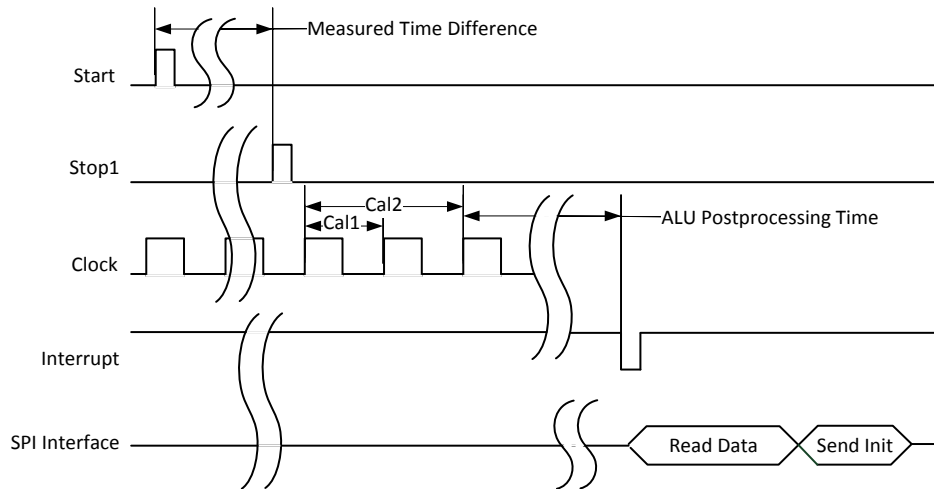


Figure 3.2 Timing – Measurement Mode 2

## Settings

Set register O, bit 11, MESSB2=1

Select measurement mode 2

Set register O, bit 13, CALIBRATE = 1

Enable calibration calculation in the ALU.

Configure Chip 2 (Measurement Mode 2):

Register	Value	Typical example configuration
Register 0	'h00042800	DIV_CLKHS = 0, the 4 MHz ceramic oscillator is internally divided by 1 START_CLKHS = 1, Oscillator continuously on CALIBRATE = 1, mandatory in measurement mode 2 to be on NO_CAL_AUTO = 0, mandatory in measurement mode 2 to have auto-calibration MESSB2 = 1, measurement mode 2 for measuring > 2 $\mu$ s. NEG_STOP/NEG_START = 0, all set to rising edges
Register 1	'h21420000	HIT2 = 1, HIT1 = 9: calculate 1.Stop CH1 – Start in measurement mode 2 EN_FAST_INIT = 0, off HITIN2 = 0, Stop CH2 disabled HITIN1 = 2, two expected hits on CH1
Register 2	'hE0000000	EN_INT = b111, interrupt given by Timeout, End Hits or ALU ready. (see also register 6) RFEDGE1 = RFEDGE2 = 0, use only rising edges
Register 3	'h00000000	
Register 4	'h20000000	
Register 5	'h00000000	CON_FIRE = b000, output configuration for pulse generator EN_STARTNOISE = 0, switch off
Register 6	'h00000000	

For further setting of the TDC-GP22, please use our data sheet.

## 4 Wide-Range Application

Some applications may wish to combine short and long range measurements. For those it is necessary to switch between measurement mode 1 (short range) and measurement mode 2 (long range).

This can be done using two TDC-GP22 devices controlled by microcontroller, one for each measurement mode. This guarantees that the range is covered with each single laser pulse.

Here, the basic schematics and general functional executable flowchart is shown.

### Basic Schematics

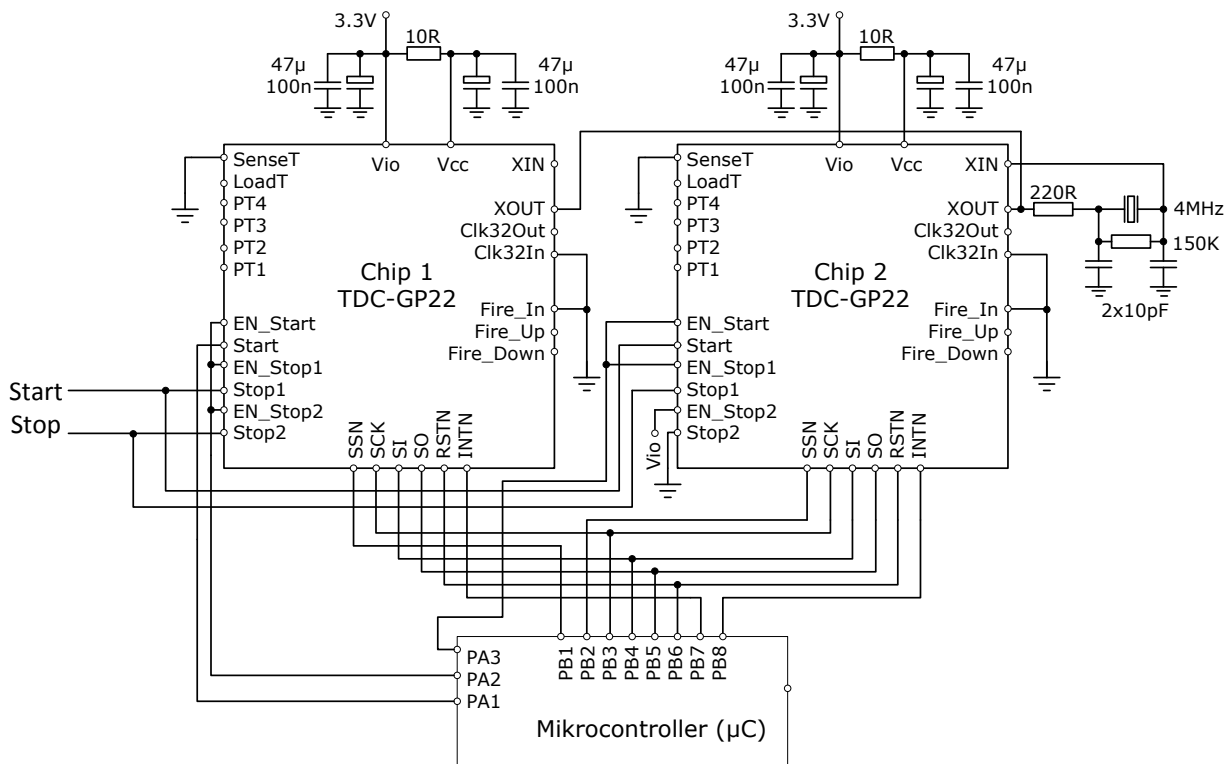


Figure 4.1 Basic Schematics



## Mikrocontroller (µC) Description

General Purpose Input/Output (GPIO)	Comment	Connected to...
Port PA1	Start-Input (generated Fake Start)	Chip 1
Port PA2	Enable Pin Start/Stop1/Stop2	Chip 1
Port PA3	Enable Pin Start/Stop1/Stop2	Chip 2
Port PB1	Slave Select (SSN)	Chip 1
Port PB2	Slave Select (SSN)	Chip 2
Port PB3	Clock Serial Interface (SCK)	Chip 1 / Chip 2
Port PB4	Data Input Serial Interface (SI)	Chip 1 / Chip 2
Port PB5	Data Output Serial Interface (SO)	Chip 1 / Chip 2
Port PB6	Reset Input (RSTN)	Chip 1 / Chip 2
Port PB7	Interrupt Flag (INTN)	Chip 1
Port PB8	Interrupt Flag (INTN)	Chip 2

## FlowChart

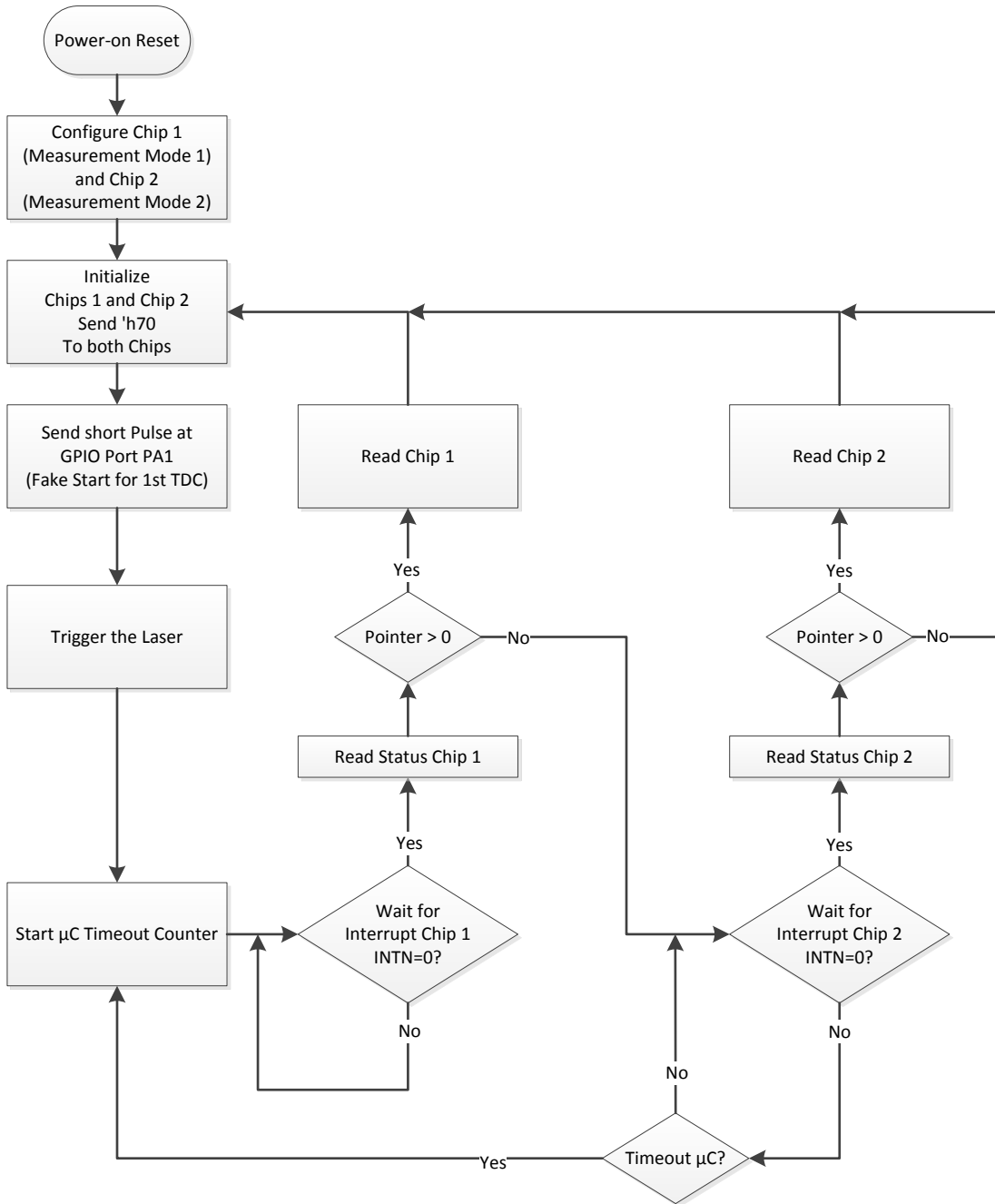


Figure 4.1 General Functional Executable FlowChart

## Settings

For further setting of the TDC-GP22, please use our data sheet.

## 5 Example

This is a generic code example for a complete laser rangefinder measurement flow including clock calibration of the high speed ceramic resonator, written for a STM32 microprocessor.

### 5.1 Detailed Flow Chart

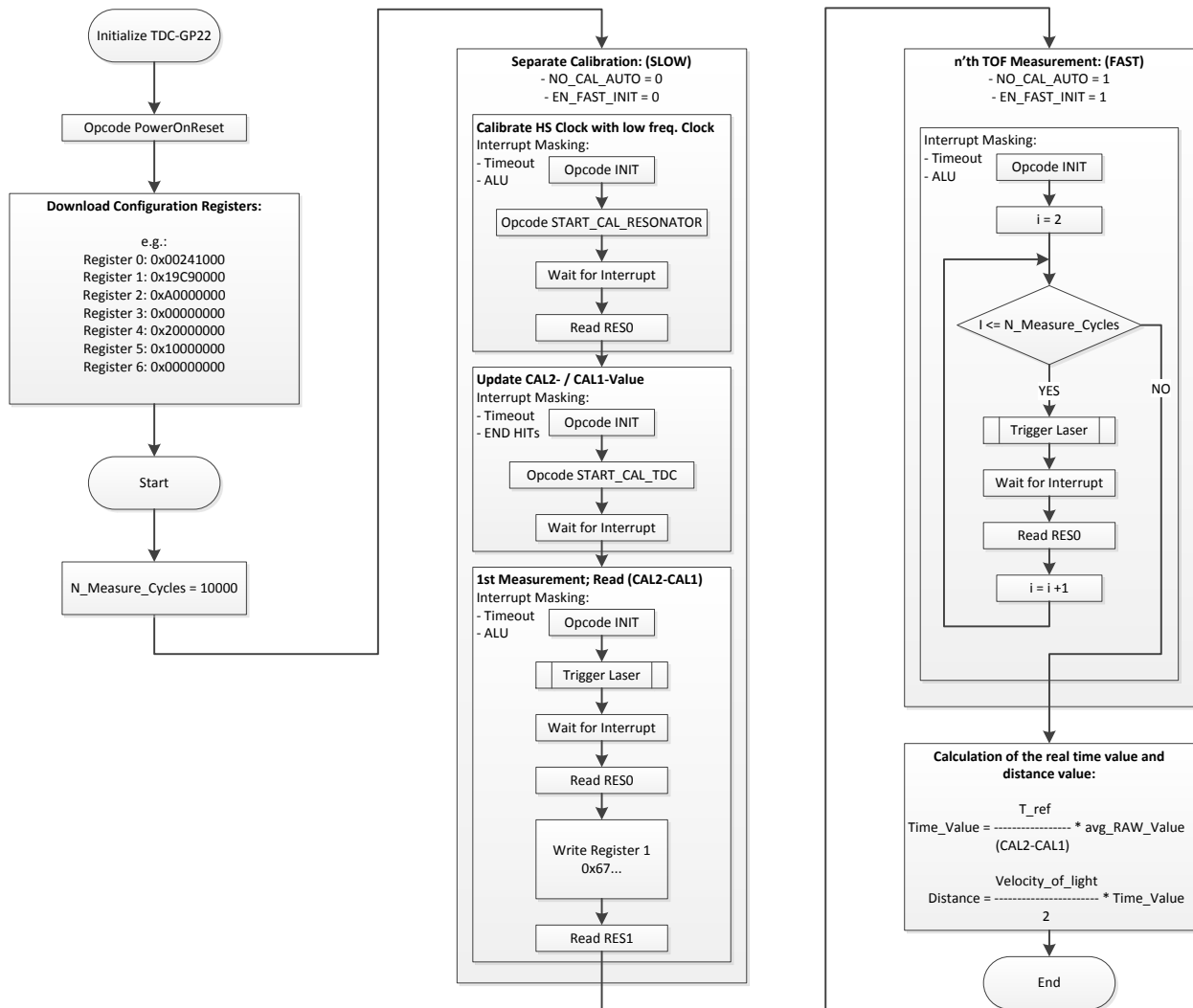


Figure 5.1 n'th Measurement with separate Calibration

## 5.2 Example Code

The whole source code can be downloaded from our download center.

```

/*****
* Function Name   : main
* Description    : Main program.
* Input         : None
* Output        : None
* Return        : None
*****/
void main(void)
{
    ENTR_CRT_SECTION();
    /* Setup STM32 system (clock, PLL and Flash configuration) */
    SystemInit();

    EXT_CRT_SECTION();

    // Choose your Slot (SPI1, SPI2)
    void* Bus_Type = SPI1;

    /* controlled loop */
    while (Dummy_var!=11) // To control the loop, e.g. (Dummy_var!=7)
    {
        if (Dummy_var==10) Dummy_var=0; // Infinite loop

        if(configured_true==FALSE)
        {
            configured_true = TRUE;
            SPIx_GPIOs_Init(Bus_Type);
            SPIx_Interface_Init(Bus_Type);
        }

#ifdef EMBEDDED_SRAM
        Embed_SRAM_Init();
#endif

        Ext_Interrupt_Init();

        gp22_send_1byte(Bus_Type, Power_On_Reset);
        Dly100us((void*)5); // 500 us wait for GP22

        // Setting of the Configuration Registers
        // CR0: DIV_CLKHS = 2, START_CLKHS = 1, CALIBRATE = 0, MESSB2 = 0, NEG_STOP =
        NEGSTART = 0, ...
        Register_0      = 0x00240000; // NO_CAL_AUTO = 0
        Register_0_NO_CAL = 0x00241000; // NO_CAL_AUTO = 1

        // CR1: HITIN2 = 1, HITIN1 = 1, ...
        Register_1      = 0x19490000; // EN_FAST_INIT = 0
        Register_1_FAST = 0x19C90000; // EN_FAST_INIT = 1

        // CR2: EN_INT, RFEDGE1 = RFEDGE2 = 0, ...
        // (NOTE: EN_INT = b111, it doesn't work with using EN_FAST_INIT)
        Register_2 = 0xA0000000; // EN_INT = Timeout(8) _ End HITs(4) _ ALU interrupt(2)

        // CR3: ...
        Register_3 = 0x00000000;

        // CR4: ...

```

```

    Register_4 = 0x20000000;

    // CR5: CON_FIRE = b000, EN_STARTNOISE = 1, ...
    Register_5 = 0x10000000;

    // CR6: QUAD_RES = 0, ...
    Register_6 = 0x00000000;

    // Writing to the configuration registers (CR)
    gp22_wr_config_reg(Bus_Type, 0x80, Register_0_NO_CAL);
    gp22_wr_config_reg(Bus_Type, 0x81, Register_1_FAST);
    gp22_wr_config_reg(Bus_Type, 0x82, Register_2);
    gp22_wr_config_reg(Bus_Type, 0x83, Register_3);
    gp22_wr_config_reg(Bus_Type, 0x84, Register_4);
    gp22_wr_config_reg(Bus_Type, 0x85, Register_5);
    gp22_wr_config_reg(Bus_Type, 0x86, Register_6);

}

// .....
// ...START_CAL_RESONATOR..Calibrate High Speed Clock.....
// ...START_CAL_TDC....Update the CAL2- and CAL1-Value.....
// .....Laser Rangefinder Measurement CYCLE.....LOOP...
// .....Caluculate Result Values.....

N_Measure_Cycles = 10000;

diff_Cal2_Cal1_old = diff_Cal2_Cal1_new;

if((Dummy_var==0) | (Dummy_var==10))
{
    //-----
    // Start Calibrate High Speed Clock Cycle (-->SLOW)
    // Important Note: NO_CAL_AUTO and EN_FAST_INIT need to be cleared!
    gp22_wr_config_reg(Bus_Type, 0x80, Register_0); // NO_CAL_AUTO = 0
    gp22_wr_config_reg(Bus_Type, 0x81, Register_1); // EN_FAST_INIT = 0

    gp22_send_1byte(Bus_Type, Init);
    gp22_send_1byte(Bus_Type, Start_Cal_Resonator);

    // Wait for INT Slot_x
    if (Bus_Type==SPI1) while (GPIO_ReadInputDataBit(GPIOD, GPIO_Pin_4)==1);
    if (Bus_Type==SPI2) while (GPIO_ReadInputDataBit(GPIOE, GPIO_Pin_11)==1);

    //Calculate Correction factor
    //The time interval to be measured is set by ANZ_PER_CALRES
    //which defines the number of periods of the 32.768 kHz clock:
    //2 periods = 61.03515625 µs
    CLKHS_freq_corr_fact = 61.03515625/
        gp22_read_n_bytes_int(Bus_Type, 2, 0xB0, 0x00) * CLKHS_freq;           // read
only two bytes

    printf("\n Correction factor for clock = %1.4f\n", CLKHS_freq_corr_fact);

    CLKHS_freq_cal = CLKHS_freq * CLKHS_freq_corr_fact; // Calibrated Clock frequency

    //-----
    // Start Separate Calibration Measurement Cycle
    // Important Note: EN_INT = End HITS
    gp22_wr_config_reg(Bus_Type, 0x82, 0x40000000); // End HITS

    gp22_send_1byte(Bus_Type, Init);

```

```

gp22_send_1byte(Bus_Type, Start_Cal_TDC); // update calibration data

// Note:
// The calibration data are not addressed directly after the calibration
// measurement but after the next regular measurement;

// Wait for INT Slot_x
if (Bus_Type==SPI1) while (GPIO_ReadInputDataBit(GPIOD, GPIO_Pin_4)==1);
if (Bus_Type==SPI2) while (GPIO_ReadInputDataBit(GPIOE, GPIO_Pin_11)==1);

// Important Note: After Separate Calibration Measurement Cycle
// EN_INT = ALU interrupt
gp22_wr_config_reg(Bus_Type, 0x82, Register_2); // Timeout + ALU interrupt

//-----
// 1st ToF Measurement plus calibration data readout
// Note: (NO_CAL_AUTO = 0 / EN_FAST_INIT = 0) --> SLOW
gp22_send_1byte(Bus_Type, Init);

//Trigger pulse laser
// SetPortHigh;
GPIO_SetBits(GPIOD, GPIO_Pin_8); // Output HIGH
// SetPortLow;
GPIO_ResetBits(GPIOD, GPIO_Pin_8); // Output LOW

// Wait for INT Slot_x
if (Bus_Type==SPI1) while (GPIO_ReadInputDataBit(GPIOD, GPIO_Pin_4)==1);
if (Bus_Type==SPI2) while (GPIO_ReadInputDataBit(GPIOE, GPIO_Pin_11)==1);

// First regular measurement (to readout calibration data)
RAW_Result_int = gp22_read_n_bytes_int(Bus_Type,2,0xB0,0x00); // read
only two bytes

#ifdef EMBEDDED_SRAM
Write_Emb_SRAM_uint32_t(RAW_Result_int); // writes the first value into SRAM
#endif

// printf("\n 1. Measured RAW Value = %u \n",RAW_Result_int); // RAW value

// Check Status Register, next free result register
// printf("Stat_Reg = 0x%04X \n",gp22_read_status_bytes(Bus_Type));

// readout the new calibration data from result register adr 0x01
gp22_wr_config_reg(Bus_Type, 0x81, 0x67490000);
diff_Cal2_Cal1_new = gp22_read_n_bytes_int(Bus_Type,2,0xB0,0x01); // read
only two bytes

#ifdef EMBEDDED_SRAM
Write_Emb_SRAM_uint32_t(diff_Cal2_Cal1_new);
#endif

}

//-----
// Calculate the real time after the hole first cycle loop
while (diff_Cal2_Cal1_old != 0)
{
    avg_diff_Cal2_Cal1 = (diff_Cal2_Cal1_new+diff_Cal2_Cal1_old) / 2;

//     printf("\n OLD Cal2-Cal1 RAW Value = %.0f \n",diff_Cal2_Cal1_old);
//     printf("\n NEW Cal2-Cal1 RAW Value = %.0f \n",diff_Cal2_Cal1_new);

```

```

    average_RAW_Result /= N_Measure_Cycles;

    // Used Formulas:
    // -----
    //           T_ref
    // Time_Value = ----- * measured_RAW_Value
    //           Cal2-Cal1
    // -----
    //           velocity_of_light
    // Distance_Value = ----- * Time_Value
    //                   2
    // -----

    // For this Source Code would be a Reference Clock used with 1 MHz

    Time_Result = (average_RAW_Result/avg_diff_Cal2_Cal1) * 1000;//time [ns]

    Distance_Result = Time_Result / 6.671281904; //distance [m]

    printf("\n Time Measure Result (ToF) = %.3f ns\n",Time_Result);
    printf(" corresponds to %.3f m of Distance\n",Distance_Result);
    printf(" to reflected point after %u Measurements\n",N_Measure_Cycles);

    diff_Cal2_Cal1_old = 0;
}

//-----
// if more than one measure cycle (-->FAST)
average_RAW_Result = RAW_Result_int; // set first value of average_RAW_Result

gp22_wr_config_reg(Bus_Type, 0x80, Register_0_NO_CAL); // NO_CAL_AUTO = 1
gp22_wr_config_reg(Bus_Type, 0x81, Register_1_FAST); // EN_FAST_INIT = 1
gp22_send_1byte(Bus_Type, Init);

//-----
// n'th ToF Measurement
for (int i=2; i<=N_Measure_Cycles;i++)
{
    //Trigger pulse laser
    // SetPortHigh;
    GPIO_SetBits(GPIOD, GPIO_Pin_8); // Output HIGH
    // SetPortLow;
    GPIO_ResetBits(GPIOD, GPIO_Pin_8); // Output LOW

    // Wait for INT Slot_x
    if (Bus_Type==SPI1) while (GPIO_ReadInputDataBit(GPIOD, GPIO_Pin_4)==1);
    if (Bus_Type==SPI2) while (GPIO_ReadInputDataBit(GPIOE, GPIO_Pin_11)==1);

    RAW_Result_int = gp22_read_n_bytes_int(Bus_Type,2,0xB0,0x00); // read
only two bytes
    // printf(" %u. Measure RAW Value = %.0f \n",i,RAW_Result_int); // RAW value
    average_RAW_Result += RAW_Result_int;

#ifdef EMBEDDED_SRAM
    Write_Emb_SRAM_uint32_t(RAW_Result_int); // writes the next values into SRAM
#endif

}

//-----

```

```
printf("\nNEW CYCLE...\n");

Dummy_var++; // To Control the loop

#ifdef EMBEDED_SRAM
    // clear internal SRAM of  $\mu$ C
    sram_mem_offset = 0x0;
#endif

} // End while Dummy_var

} //End main
```



## **6 Miscellaneous**

### **6.1 References**

**EECS University of Central Florida – Department of Electrical Engineering and Computer Science**

<http://www.eecs.ucf.edu>

Project: “Portable Sensing Field Sensor”, Group-No. 4 (2011)

<http://www.eecs.ucf.edu/seniordesign/sp2011su2011/g04/index.html>

Project: “Red Eye”, Group-No. 8 (2011)

<http://www.eecs.ucf.edu/seniordesign/su2011fa2011/g08/index.html>

**ECE ILLINOIS University of Illinois – Department of Electrical and Computer**

<http://www.ece.illinois.edu/>

Project “1D LiDAR TOF Rangefinder”, Group-No.28 (2012)

<https://courses.engr.illinois.edu/ece445/?f=Projects&sem=spring2012&proj=28#a28>

Project “Laser Range Finder”, Group-No. 3 (2007)

<https://courses.engr.illinois.edu/ece445/?f=Projects&sem=summer2007&proj=3#a3>

## 6.2 Literature Guide

### Data Sheets

Title	Document-No
TDC-GP22 Universal 2-Channel Time-to-Digital Converter dedicated to Ultrasonic Heat & Water Meters	DB_GP22_x.pdf

The latest version of the available document can be downloaded from the acam website

<http://www.acam.de/download-center/time-to-digital-converters/>

## 6.3 Document History

27.02.2013	First release
25.03.2013	Version 1.1 for release, page 14, Basic schematics revised
08.01.2014	Version 1.2 for release, section 7 Example Code including part "Calibrate High Speed Clock" and EN_FAST_INIT modified,
28.01.2014	Version 1.3 for release, section 2.1, Register 2 revised; Detailed flow chart of example source code added; Example source code regarding timing optimized

AN034



acam-messelectronic gmbh  
Friedrich-List-Straße 4  
76297 Stutensee-Blankenloch  
Germany  
Phone +49 7244 7419 - 0  
Fax +49 7244 7419 - 29  
E-Mail [support@acam.de](mailto:support@acam.de)  
[www.acam.de](http://www.acam.de)