Sensors & Transducers, Vol. 271, Issue 4, December 2025, pp. 58-70

Sensors & Transducers
Published by IFSA Publishing, S. L., 2025
http://www.sensorsportal.com

A High-resolution Time-to-digital Converter with a Time-stretching Technique in a General-purpose Microcontroller

[1] Geu M. PUENTES-CONDE, [1, *] Ernesto SIFUENTES, [1, *] Javier MOLINA, [1] Francisco ENRÍQUEZ-AGUILERA, [2] Abraham MEJÍA-AGUILAR and [3] Guadalupe NAVARRO ENRÍQUEZ
[1] Institute of Engineering and Technology, Universidad Autónoma de Ciudad Juárez (UACJ),
Ciudad Juárez 32310, México
[2] Eurac Research, Center for Sensing Solutions, Viale Druso 1, 39100, Bolzano, Italy
[3] Tecnológico Nacional de México, Instituto Tecnológico de Ciudad Juárez (ITCJ),
Ciudad Juárez 32500, México
Tel.: +52-656-688-4841
[*] E-mail: esifuent@uacj.mx, jamolina@uacj.mx

Received: 16 October 2025 / Revised: 23 Nov. 2025 / Accepted: 24 Dec. 2025 / Published: 30 Dec. 2025

Abstract: Time-to-digital converters (TDCs) are electronic devices used in systems that require precise measurement of time intervals. These devices transform temporal information into digital form. Traditionally, the resolution of TDCs built on general-purpose microcontrollers (MCUs) has been limited by the reference clock frequency, which typically does not exceed 80 MHz. Recent advances in quasi-digital techniques offer practical solutions to overcome this limitation, primarily through time-stretching and analog interpolation. This article presents a new high-resolution TDC architecture that incorporates a time-stretching technique within a 16-bit MCU and two analog interpolators, implemented via direct-interface circuits. A time interval resolution of 4 ns was achieved over a range of 10 µs to 16 ms using a 4 MHz reference clock. This performance is comparable to that of a 240 MHz digital counter, representing a sixty-fold improvement in effective clock resolution. The proposed architecture is well-suited for portable, low-power applications that do not compromise measurement accuracy.

Keywords: Time-to-digital converter, Analog interpolator, Time-stretching, Direct interface circuit, Microcontroller Unit (MCU).

---

1. Introduction

A time-to-digital converter (TDC) is an electronic device that converts temporal information into a digital form in systems that require measuring the elapsed time between two events. Fig. 1 depicts a block diagram of a TDC, where the time interval of interest, T_x, is encoded from the time domain into a digital output, D_{out}, for subsequent processing.

Applications of TDCs include electronic instrumentation [1], Time-of-Flight (ToF) systems such as laser radar [2, 3], Light Detection and Ranging (LiDAR) [4], distance measurement [5], spectrometry [6–8], particle detection [9], and on-chip sensing [10–12], among others.

Henzler [13] described traditional TDC as a simple counter that measures time intervals by counting the cycles of a reference clock; however, the resolution of such a counter is limited by the frequency of the reference clock, which is constrained by hardware capabilities (e.g., 300–500 MHz in commercial FPGAs or approximately 80 MHz in most MCUs).

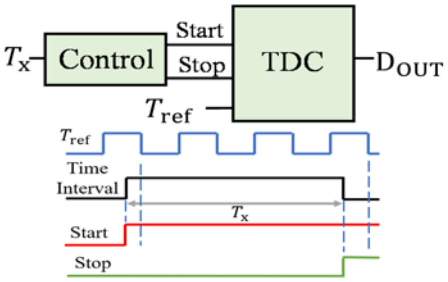

Fig. 1. TDC Block diagram and its time interval diagram.

Although higher clock frequencies improve TDC resolution and reduce quantization errors in the start ($\Delta T_{\mathrm{start}}$) and stop ($\Delta T_{\mathrm{stop}}$) triggers, they also increase power consumption.

TDCs share characteristics with Analog-to-digital converters (ADCs); the key difference lies in measuring a continuous voltage by an ADC versus discrete time intervals by a TDC. In ADCs, the input voltage is compared to a reference V_{ref}, which defines the maximum level without saturation. This reference is typically supplied externally, yielding a least-significant-bit (LSB) voltage defined as $V_{\mathrm{LSB}} = V_{\mathrm{ref}}/2^N$, where N denotes the device's resolution in bits. In contrast, TDCs lack an explicit reference time, T_{ref}; instead, a quantization time, T_{LSB}, is defined as the smallest detectable time increment. Consequently, the maximum measurable time span of a TDC can be expressed as $T_{\mathrm{ref}} = 2^N \times T_{\mathrm{LSB}}$. These parameters illustrate the relationships among reference time, time resolution, and bit depth in TDC design.

TDC architectures are classified into generations based on their underlying techniques [13–15]. First-generation TDCs employed quasi-digital approaches, such as converting time intervals into an amplitude (e.g., time-to-voltage converters [5, 16, 17]) or using single-, dual-, and triple-slope integration followed by digital counting, with resolution enhanced through time-stretching and interpolation methods [18–22]. Second-generation TDCs utilized direct measurement techniques by comparing time intervals against known reference values, which are commonly implemented using delay lines [23, 24], Vernier delay lines [25–27], or pulse-shrinking lines [28, 29].

The increasing demand for compact, energy-efficient systems calls for high-resolution, precise time-measurement solutions. For example, Bengtsson proposed a low-cost, high-resolution TDC based on quasi-digital techniques implemented with an MCU [30]. Additionally, time-stretching is achieved through a direct-interface circuit that uses a capacitor for charging and discharging, enabling accurate time interval estimation [31, 32].

Later, a Vernier method was implemented using two oscillators with slightly differing frequencies, whose pulse synchronization yielded time-interval estimation [33].

This article proposes another TDC approach using a mid-range MCU and a few electronic components as direct interface circuits [31, 32], time-stretching [30], and interpolation techniques [20-22] to mitigate uncertainties in the start and stop trigger signals caused by reference clock limitations, achieving high resolution with low complexity and low cost.

2. Materials and Methods

This article presents a hardware circuit that reduces time quantization error in the embedded timer of a general-purpose MCU, which is affected by the reference clock's resolution. To achieve this, Fig. 2 shows the block diagram of the hardware circuit, organized into three main modules:

1. An analog interpolator: Enhances the resolution by capturing fine details of the start and stop trigger signals of an entire time interval;
2. A Time-to-Voltage Converter: Converts the time interval uncertainties into a proportional voltage;
3. A time-stretching circuit: Expands the time interval for precise quantization, effectively reducing the uncertainty at start and stop triggering signal errors.

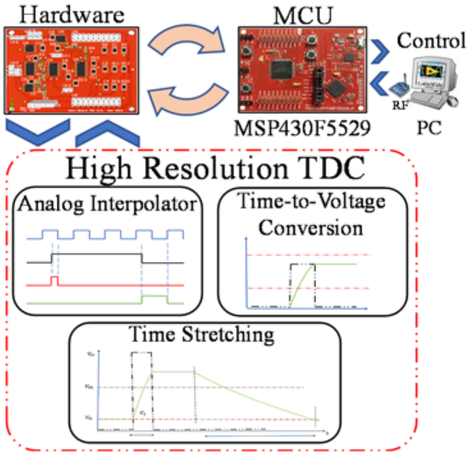

Fig. 2. Block diagram of the proposed circuit.

These modules provide a complete implementation on a printed circuit board (PCB) that is directly connected to the Texas Instruments development board with an MCU MSP430F5529. Additionally, RF communication with a computer (PC) is achieved using an XBee module transceiver, specifically the XB24CZ7WIT-004.

The following sections describe the hardware circuit and its main characteristics for each module.

2.1. Analog Interpolator

When measuring a time interval T_x with the MCU's embedded digital timer, a quantization error, commonly known as uncertainty, is introduced when detecting the start and stop trigger signals (T_{Start} and T_{Stop}), as shown in Fig. 3.

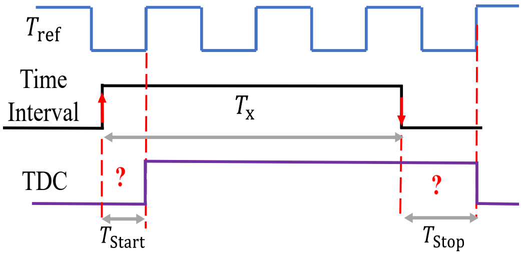

Fig. 3. Quantization error in a time interval measurement with a digital timer.

This uncertainty is due to synchronization errors between the reference clock T_{ref} and the digital timer during T_x detection. These discrepancies can result in inaccuracies in timing measurements.

An improvement to the 8-bit MCU-based TDC by Bengtsson [30], which corrects uncertainties at T_{Start} and T_{Stop}, is presented in Fig. 4. The timing diagram shows the proposed analog interpolator, a modification of the technique described in [17-19]. The aim is to capture the uncertainty into two asynchronous signals by synchronizing the rising and falling edges to estimate T_{Start} and T_{Stop}. Thus, it allows assessing the synchronized time interval T_s based on the number of counts N and the MCU's reference clock T_{ref}: $T_s = T_{\text{ref}} \times N$.

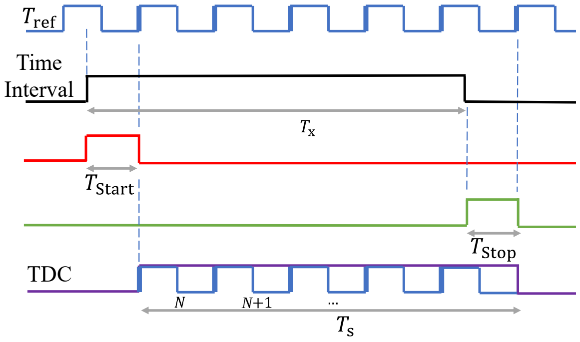

Fig. 4. Timing diagram of the analog interpolator.

The analog interpolator obtains the T_{Start} and T_{Stop} signals by modifying the electrical circuit, reducing the number of two-D-type flip-flop chains required to capture the start trigger signal, T_{Start}, and the stop trigger signal, T_{Stop}. In other words, the circuit is simplified by eliminating four flip-flops and removing an EXOR gate from the traditional design.

Fig. 5 shows the electrical circuit and its operational stages for obtaining T_{Start} using fewer components, comprising a D-type flip-flop, a buffer inverter, and an OR gate that interact with the MCU reference clock.

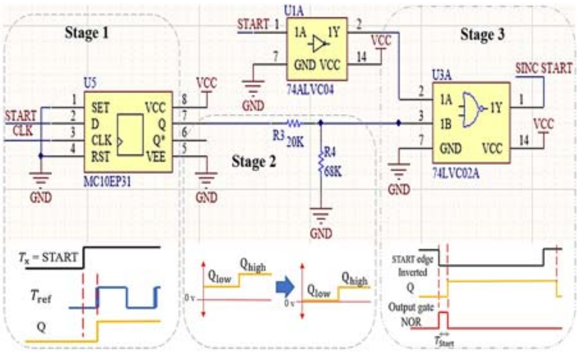

Fig. 5. Electrical circuit for capturing the start trigger signal.

- **Stage 1.** A D-type flip-flop is used to synchronize the input of T_x (identified as START) with an input T_{ref} from the MCU reference clock at 4 MHz;
- **Stage 2.** A voltage divider is applied to the D-type flip-flop due to its architecture, which operates in Positive Emitter-Coupled Logic (PECL) mode, with logical state transitions in Q at 25 °C occurring at $V_{\text{oLow}} = 1.55$ V and $V_{\text{oHigh}} = 2.35$ V. Therefore, V_{oLow} is adjusted close to 0 V;
- **Stage 3.** T_{Start} is captured with an inverter OR gate to detect the start trigger signal of T_x and compares it with the signal coming from the flip-flop (input 1B) to extract the small-time fragment T_{Start}.

To obtain T_{Stop}, the same components used for the T_{Start} circuit are employed, with only one additional element at stage 1: an inverter buffer gate. This gate inverts the level of the stop-edge T_x signal to work with a rising-edge signal. From this point forward, the following stages are the same as outlined for T_{Start}.

2.2. Time-to-voltage Conversion

Time-to-Voltage conversion translates minor uncertainties of the time intervals T_{Start} and T_{Stop} into the amplitude domain. Fig. 6 shows the primary schematic, in which the conversion is performed by charging and discharging a reference capacitor, as described in direct interface circuits [31, 32].

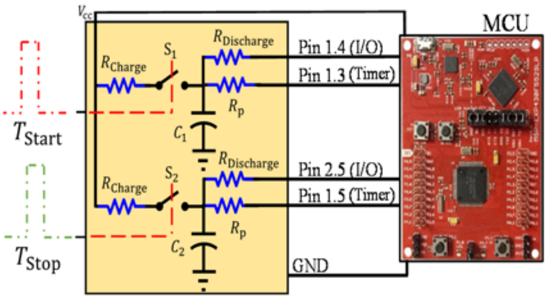

Fig. 6. General representation of the proposed time-to-voltage converter.

The charging process is performed through analog switches S_1 and S_2, which switch charging through a current-limiting resistor R_{Charge} and interact with the MCU to initiate a time-stretching process with a current-limiting discharge resistor, an improved adaptation of the initial stretching technique proposed in [30]. C_1 and C_2 are charged to V_{cc} with respect to the time period of T_{Start} and T_{Stop}. The transient voltage response across the capacitors is shown via the exponential function described in Eq. (1).

$$V_c(T_{\text{Start}}) = V_{cc}(1 - e^{-\frac{T_{\text{Start}}}{RC}}),$$
$$V_c(T_{\text{Stop}}) = V_{cc}(1 - e^{-\frac{T_{\text{Stop}}}{RC}}) \tag{1}$$

The MCU embedded timer operates in V_{TL} and V_{TH}. If a time interval in the amplitude domain falls below V_{TL}, as shown in Fig. 7, capturing the capacitor's discharging time becomes impractical in terms of capacitor voltage charging.

Therefore, the time constant of an RC network must satisfy the thresholds of the MCU's Schmitt trigger; this means that the capacitor voltage charge (V_{Charge}) must be greater than V_{TL}, and the capacitor voltage charge must not saturate ($0.99V_{cc}$) to avoid losing information, as shown in Eq. (2).

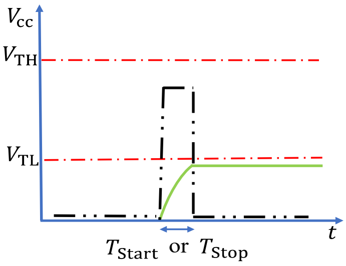

Fig. 7. Voltage below the threshold V_{TL}.

$$V_{\text{TL}} < V_{\text{Charge}} < 0.99V_{cc} \tag{2}$$

To address this issue, a software-based offset voltage is developed to correct the uncertainty in both capacitors associated with the start and stop trigger signals. This ensures that the capacitor remains charged to the V_{TL} voltage. Fig. 8 shows the charging process for the stretching capacitor, $C_{stretch}$.

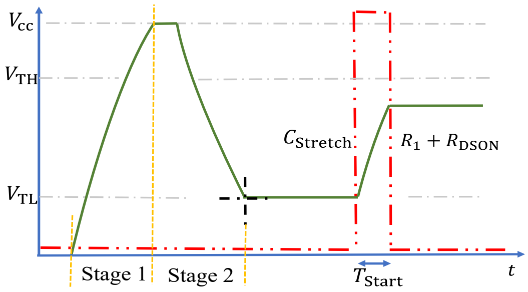

Fig. 8. Offset voltage to V_{TL} of the stretching capacitor, and a start trigger signal T_{Start}.

In stage 1, the voltage offset is established from 0 V to V_{cc}, and in stage 2, the capacitor discharges to capture the low voltage threshold V_{TL} range of the MCU, before the arrival of a time interval.

Additionally, to satisfy Eq. (2) by ensuring the capacitor charge does not exceed 99 % saturation. The charging characteristics of the stretching capacitors are governed by the current-limiting resistances $R_1 + R_{DSON}$, which is an external current-limiting resistor in series with the R_{DSON} resistor of the analog switch. Fig. 9 shows the electrical circuit for both stretching capacitors: one for the start trigger signal and one for the stop trigger signal.

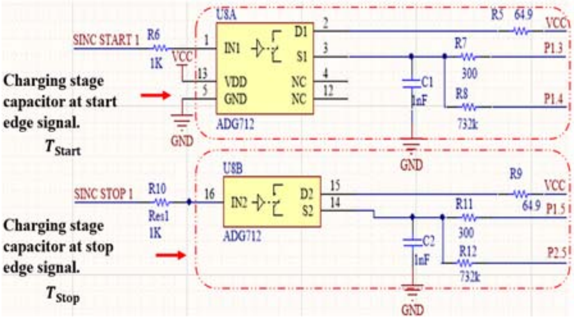

Fig. 9. The electrical circuit of the stretching capacitors at start and stop trigger signals.

Each capacitor's charge is limited to one MCU clock cycle (not exceeding 250 ns) plus an offset to reach the threshold voltage V_{TL} as shown in Eq. (3), which is delimited based on the capacitor charge voltage (V_{Charge}) in terms of time (t) at the $R_1C_{stretch}$ network by rearranging Eq. (2).

$$-R_1C_{\text{stretch}} \ln\left[1 - \frac{V_{\text{TL}}}{V_{\text{CC}}}\right] < t < $$
$$< 4.606(R_1C_{\text{stretch}}) \tag{3}$$

$R_1C_{stretch}$ network is directly proportional to the thresholds described by Eq. (2), specifically the values of the current-limiting resistance R_1 and the stretching capacitor $C_{stretch}$ represents that meet the timing constraints of Eq. (3), which are derived by rearranging in terms of the ratio of $R_1C_{stretch}$ to meet the values needed for the passive components as shown in Eq. (4) to ensure that the capacitor voltage charge must not be saturated to avoid losing information.

$$\frac{-t}{\ln\left[1 - \frac{V_{\text{TL}}}{V_{\text{CC}}}\right]} < R_1C_{\text{stretch}}. < 0.2171(t) \tag{4}$$

2.3. Time-stretching

The time stretching module is related to the V_{TL} and V_{TH} thresholds of the MCU's embedded timer, which acts as comparators that detect the rising and falling edges of a time interval to initialize, reset, or stop the timer. Time stretching is initiated through software by manipulating the MCU's pins to start timers in capture mode, as shown in Fig. 10.

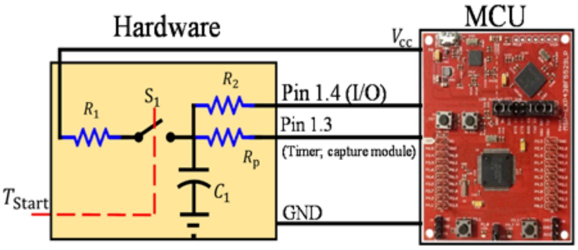

Fig. 10. Voltage-to-digital conversion of T_{Start} To a time-stretching process.

Fig. 11 shows the timing diagram for the charge/discharge process of both stretchable capacitors, which is used to determine the uncertainty of a time interval based on a given stretch factor. This uncertainty is used to satisfy the time-stretching requirement for determining the T_{Start} and T_{Stop}.

The primary function of the discharge capacitor is to capture the information encoded in the amplitude domain and to process it into a digital value by the MCU, which means stretching the time interval by a factor of k.

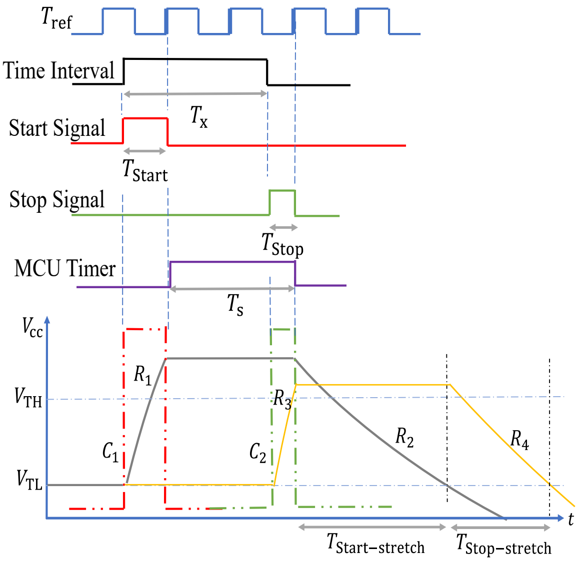

Fig. 11. Timing process of time stretching for T_{Start} and T_{Stop}.

Eq. (5) is obtained to determine the stretching factor (k) of the voltage in the capacitor C_1 for a time interval T_{Start}, given a slow discharge time (t_d) through the current-limiting resistor R_2, and a charging time through the current-limiting resistor R_1. Note that the discharging time is defined by $t_d = R_2C_1 \cdot \ln(V_I/V_{TL})$.

$$k_1 = \frac{t_d}{t} = -\frac{R_2 C}{t}\left\{\ln\left[\frac{V_{TL}}{V_{CC}\left(1-e^{-t/R_1 C}\right)}\right]\right\} \tag{5}$$

The stretching factor of the capacitor C_2 for a time interval T_{Stop}, given a slow discharge time (t_d) is obtained using Eq. (5) in terms of R_4C_2.

To calculate the current-limiting resistances R_2 and R_4, it is necessary to consider the maximum time interval t_{dmax} that the MCU's embedded timer can measure. This corresponds to a frequency of 4 MHz, with a period of 250 ns, in a 16-bit embedded timer, allowing 65,535 counts. Therefore, t_{dmax} = 16.38 ms. By establishing the time limits for the discharge of the stretchable capacitor, the value of the current-limiting resistance can be directly obtained from Eq. (5) to achieve the total margin of counts, as shown in Eq. (6).

$$R_2 = R_4 = k\frac{t_{dmax}}{\ln\frac{V_{TL}}{V_{CC}}(C)} \tag{6}$$

2.4. Linear Interpolation

The charging and discharging processes of a capacitor exhibit nonlinear behavior; the voltage across the capacitor's terminals over time, as expressed in Eq. (1), follows an inverse exponential (logarithmic) function during the charging phase. In contrast, the voltage during the discharging phase over time $V_c(t)$ described in Eq. (7), follows an exponential function, as shown in the timing diagram in Fig. 12.

$$V_c(t) = V_{cc}(e^{-\frac{t}{RC}}) \tag{7}$$

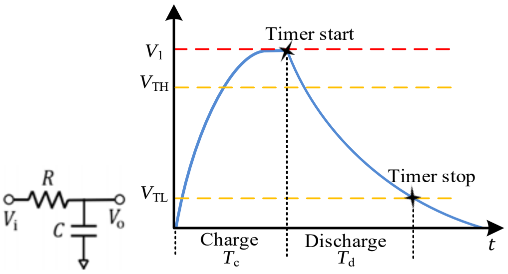

Fig. 12. Behavior of charging and discharging capacitors throughout an RC network.

A solution to correct and estimate values is to correlate the input data with the TDC’s output response by calibrating the system with a known reference instrument. Experimental data points are stored (x_1,y_1) and (x_2,y_2) as reference values for the lookup table, and software is used to search in the lookup table the corresponding values across the entire range considered, ensuring accurate interpolation using Eq. (8). During operation, the system references the lookup table to find the closest matching value (y), enabling precise linearization via software and correction of the TDC's output response.

$$y = y_1 + \frac{x - x_1}{x_2 - x_1} \cdot (y_2 - y_1) \qquad (8)$$

2.5. Time-to-digital Conversion

The proposed circuit shown in the diagram in Fig. 2 operates across four functional measurement stages: (i) initialization of the time measurement system, (ii) measurement of the time interval synchronized to the MCU reference clock (T_s), (iii) measurement of the time stretching value of T_{Start}, and (iv) measurement of the time stretching value of T_{Stop}. Eq. (9) represents the sum of the three measurements, which together reduce the uncertainty in measuring the entire time interval (T_x) as shown in Fig. 13.

$$T_x = T_s + (T_{Start} - T_{Stop}) \qquad (9)$$

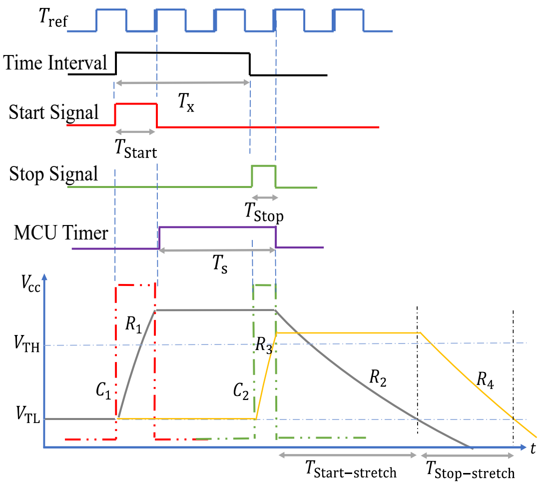

Fig. 13. Time capture for a time interval T_x.

The TDC for T_{Start} and T_{Stop} is achieved by solving Eq. (5) for $t_{dmax} = N \times T_{ref}$ and determining the time at which the stretchable capacitor was charged, based on its time stretching as shown in Eq. (10).

$$[N][T_{ref}] = -R_2 C \left\{ \ln \left[\frac{V_{TL}}{V_{CC}\left(1 - e^{-t/R_1 C}\right)} \right] \right\} \qquad (10)$$

Solving Eq. (10) in terms of time, its digital expression for T_{Start} (or T_{Stop}) can be expressed as

$$T_{Start} = -R_1 C \ln \left[1 - e^{\frac{[N][T_{ref}]}{R_2 C} + \ln\left[\frac{V_{TL}}{V_{CC}}\right]} \right] \qquad (11)$$

The process of obtaining time intervals (T_s, T_{Start}, and T_{Stop}) involves using three timers on the MCU, each performing a specific task in a specified order:

1. System initialization system using Timer A2 on pin 2.4, waiting for an interrupt service in capture mode on the rising edge (capturing of T_{Start});
2. Time interval (T_s) synchronized to the reference clock T_{ref} using Timer A2 on pin 2.4 in capture mode on the falling edge (capturing of T_{Stop});
3. Time stretching due to the uncertainty of the start trigger signal T_{Start} using Timer A0 on pin 1.3 in capture mode on the falling edge;
4. Time stretching due to the uncertainty of the stop trigger signal T_{Stop} using Timer A2 on pin 1.5 in capture mode on the falling edge.

3. Results

Results are based on the validation and characterization of the implemented novel high-resolution TDC hardware circuit, as shown in Fig. 14.

Fig. 14. Hardware circuit.

According to the block diagram in Fig. 15, the MCU serves as the system's brain, implementing the conversion algorithms and managing interactions among the hardware modules.

A Keysight 33500B function generator was used to generate a series of time intervals. Through a wireless transceiver (X-Bee), the MCU communicates with a LabVIEW program to control functionality, visualize measurements, and record data.

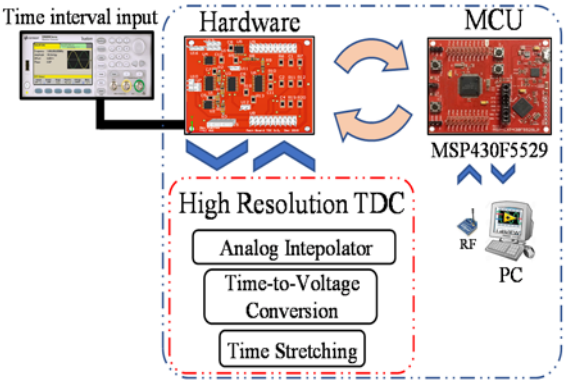

Fig. 15. Time interval measurement validation.

3.1. Analog Interpolator

The analog interpolator circuit shown in Fig. 16 is characterized using a Keysight DSOX3014A oscilloscope with a 2 µs time interval, T_x, supplied by the Keysight 33500B function generator.

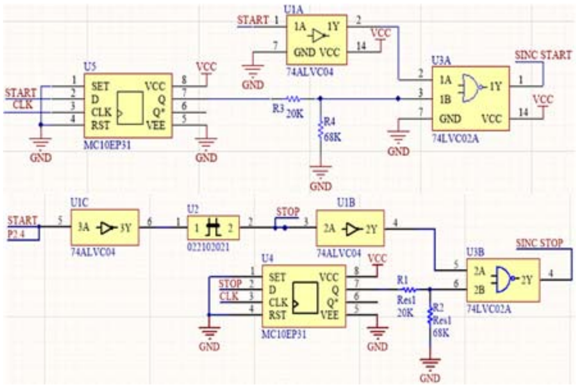

Fig. 16. Electrical circuit of the analog interpolator.

This interval is compared to the reference clock T_{ref}, sourced from the MCU's 4 MHz clock. The interpolator captures both the start T_{Start} and stop T_{Stop} trigger timers, which are shorter than one reference clock cycle, as shown in Fig. 17.

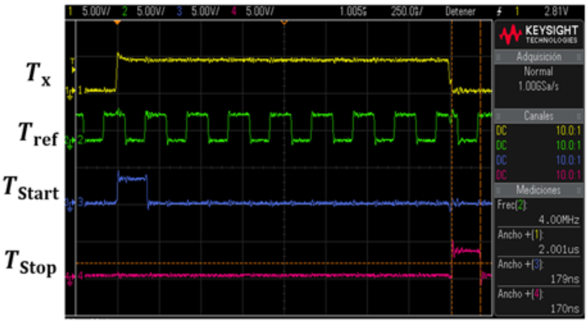

Fig. 17. Experimental results of the analog interpolator for a 2 µs time interval.

3.2. Time-to-voltage Conversion

The time-to-voltage conversion stage was crucial for converting time-domain information into amplitude-domain information. Therefore, working conditions were established to prevent information loss during the conversion process. The V_{TL} level in the MCU was adjusted to capture the capacitor charge and discharge using an oscilloscope; Fig. 18 shows the V_{TL} being controlled by software.

The measured threshold was close to $V_{\mathrm{TL}} = 1.2875$ V, and it was stored in a dynamic variable to capture the start and stop trigger times as digital values, ensuring that the capacitor voltage charge satisfies $V_{\mathrm{TL}} < V_{\mathrm{Charge}} < 0.99\ V_{\mathrm{cc}}$ from the MCU's Schmitt trigger.

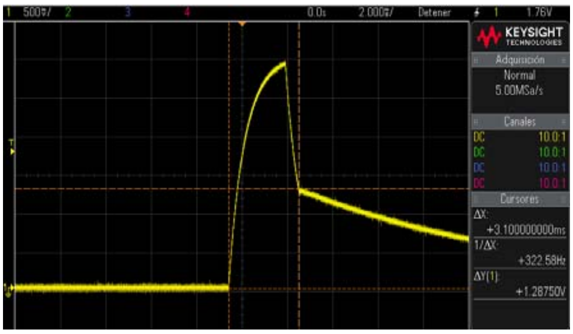

Fig. 18. Capture of V_{TL} with the oscilloscope.

The capacitor's charging is directly related to the time constant R_1C_{stretch}. In terms of time, Eq. (4) is used to establish the limits for converting time to voltage in a capacitor and to ensure a maximum charge at 5*RC*.

Using a Keysight 34450A 5 ½-digit digital multimeter, Table 1 presents the measured values of R_1, R_{DSON}, and C_{stretch}, along with the estimated charging times.

Table 1. *RC* network practical values and time interval ranges.

Time	Ranges	
T_{Start}	$R_{1\ \mathrm{Start}} + R_{\mathrm{DSON}}$	$C_{\mathrm{Start-stretch}}$
	64.9 Ω + 2.50019 Ω	1.0314 nF
	$4.22\ \mathrm{ns} < T_{\mathrm{Start}} < 321.14\ \mathrm{ns}$	
T_{Stop}	$R_{1\ \mathrm{Stop}} + R_{\mathrm{DSON}}$	$C_{\mathrm{Stop-stretch}}$
	64.9 Ω + 2.50019 Ω	1.0197 nF
	$34.05\ \mathrm{ns} < T_{\mathrm{Stop}} < 317.17\ \mathrm{ns}$	

The time required to reach the V_{TL} level in both *RC* circuits is approximately 34 ns, and this correction shifts the voltage from 0 V to $V_{\mathrm{TL}} = 1.2875$ V, allowing the remaining time interval to reach its maximum charge. Specifically, for the *RC* networks at $T_{\mathrm{Start}} = 286.92$ ns, and = 283.12 ns. It is sufficient to detect any uncertainties below one clock cycle on the MSP430F5529 MCU running at 4 MHz. Therefore, we can conclude that the *RC* network can capture uncertainties below 250 ns without saturating $V_{\mathrm{Charge}} < 0.99\ V_{\mathrm{cc}}$.

Fig. 19 presents the uncertainty of a start trigger time T_{Start} from a time interval T_x with a pulse width of 2 µs to the reference clock T_{ref}.

Time-to-voltage conversion was performed based on the pulse width of the start trigger's uncertainty, $V_{\mathrm{Charge}}(T_{\mathrm{Start}})$. It was observed that the capacitor voltage is approximately 2.7 V for a charging time of $T_{\mathrm{Start}} = 120$ ns. Therefore, the voltage across the capacitor, as stated in Eq. (8), is $V_{\mathrm{Charge}}(t) = 2.7097$ V.

Similarly, Fig. 20 shows the capture of uncertainty in the stop trigger time, T_{Stop}, of the time interval T_x, followed by its conversion to voltage, $V_{\text{Charge}}(T_{\text{Stop}})$.

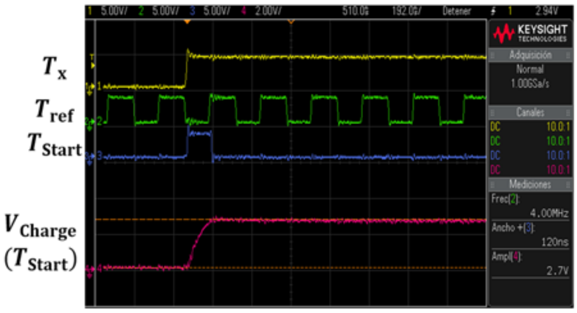

Fig. 19. Time-to-voltage conversion of the start trigger uncertainty.

Fig. 20. Time-to-voltage conversion of the stop trigger uncertainty.

The charge level in capacitor C is approximately 2.9 V for a charging time of 139 ns, as indicated by T_{Stop}. Thus, according to Eq. (8), the voltage across the capacitor with respect to time is $V_{\text{Charge}}(t) = 2.86$ V.

3.3. Time-stretching

Time-stretching is performed through software. The capacitor's discharges, $C_{\text{Start-stretch}}$ and $C_{\text{Stop-stretch}}$, are initialized so that the time stretch is long enough to be detected by the MCU timers. A maximum time stretch of no more than 650 µs is established for a maximum time of 250 ns, resulting in a stretching factor of $k = t_d$ / t = (650 µs) / (250 ns) = 2600. The discharge time in the capacitors, denoted as t_d, is the duration required to measure a specific time interval.

The discharge resistance values were selected using Eq. (7) and are shown in Table 2.

Therefore, commercial resistance values of 732 kΩ were chosen and measured with the digital multimeter, yielding R_{dStart} = 731.47 kΩ and R_{dStop} = 731.22 kΩ.

These values resulted in a maximum time stretching of 690 µs, with both uncertainties at 250 ns. Fig. 21 shows the behavior seen at both stretching capacitors, which was analyzed using 50 samples taken over a time interval defined by the Keysight 33500B function generator in LabVIEW.

Table 2. Discharge capacitor stretching resistor value for start/stop trigger signal.

Time	T_{Start}	T_{Stop}
	$R_{\text{dStart}} = 687.778\ \Omega$	$R_{\text{dStop}} = 695.676\ \Omega$

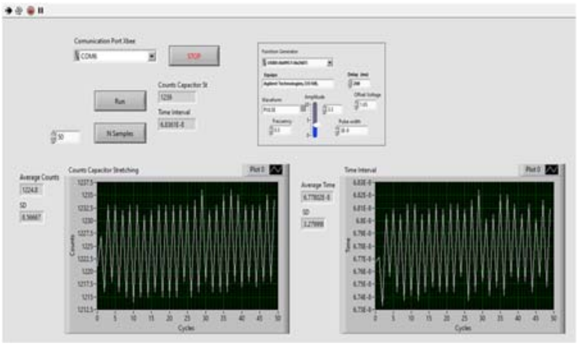

Fig. 21. Graphical interface in LabVIEW to control time-stretching periods.

Fig. 22 shows the trend curve of the time-stretch characterization obtained during the capture of uncertainty in the start and stop triggers. Thus, we can graphically observe the inverse exponential (logarithmic) trend in the response to the measurements, similar to the charging of a capacitor.

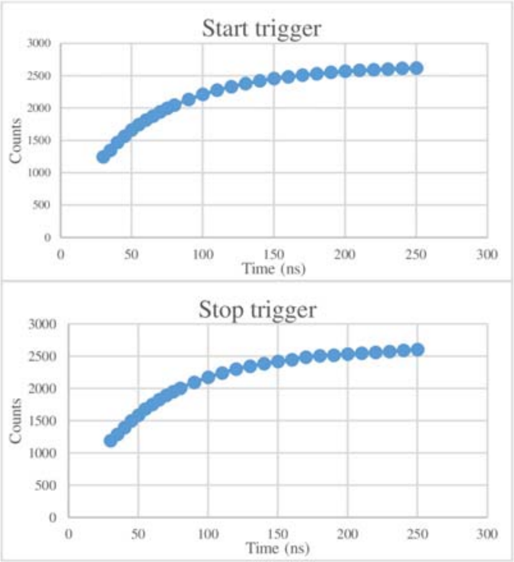

Fig. 22. Characteristic curve of uncertainty capture for the start and stop trigger signal (T_{Start} and T_{Stop}).

Results were compared between the average counts per time of the MCU's timer and its standard deviation, calculated to analyze its stability and characteristic curve over specific time intervals.

Note: Practically, the characterization of the time stretching for different periods in T_{Start} or T_{Stop} was obtained using stimuli ranging from 30 ns to 250 ns.

The reason for starting at 30 ns was the function generator's stability, which cannot produce square waves shorter than 30 ns with good stability.

Regarding the uncertainty of the start trigger, it was observed that at smaller time intervals, specifically at 30 ns, the maximum tendency to vary was below 13.33 counts. In contrast, at larger time intervals up to 250 ns, the minimum tendency was below 0.91538 counts. Due to the uncertainty of the stop trigger, it was observed that the maximum tendency was below 15.6 counts. The minimum tendency to vary was below 3.4 counts.

To analyze the voltage-to-time conversion for the start and stop trigger, 50 samples were obtained at consistent time intervals, and the time for each trigger was estimated, specifically for T_{Start} and T_{Stop}—directly from Eq. (12). Fig. 23 shows the graphs of the results obtained for T_{Start} and T_{Stop}, featuring a linear trend line fitted using the least squares method in Microsoft Excel, which achieved coefficients of determination of up to 0.9994 and 0.9996 applying the linear interpolation technique with lookup tables.

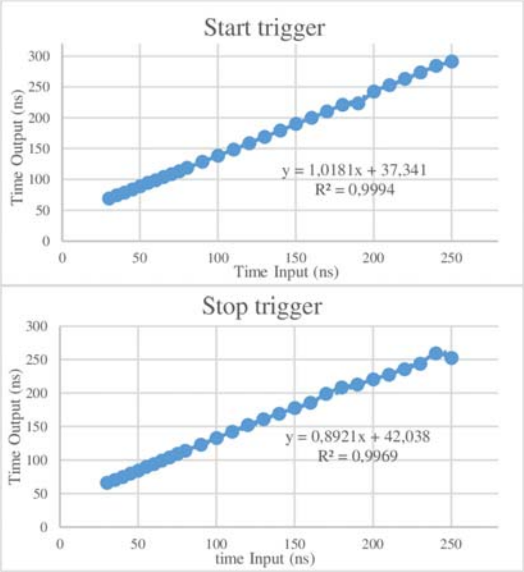

Fig. 23. Graph of the start and stop trigger's input time vs output time.

The maximum variability observed for the start trigger was 2.02 ns at a 250 ns interval, while the minimum was 0.4469 ns at a 50 ns interval. On the other hand, the maximum variability captured for the stop trigger was 4.17 ns at a time interval of 250 ns. In comparison, the minimum variability recorded was 0.1156 ns over a 30 ns time interval. Measurements were obtained from the time-stretching process. Fig. 24 shows the stretching behavior, characterizing the uncertainty in the start and stop triggers across different time intervals, expressed as $k = t_d/t$.

Linear interpolation based on the data from the start and stop trigger uncertainty characterization calibrated the TDC's time-stretching response using a lookup table. Initially, the measured and ideal time ranges were mapped to a 216-count margin, allowing them to be processed as a lookup table containing 28 ideal input time data points for linear interpolation.

Consequently, the system was linearly scaled from 0 counts to 65535 counts, considering potential time values ranging from 0 to 250 ns, including both measured and ideal values.

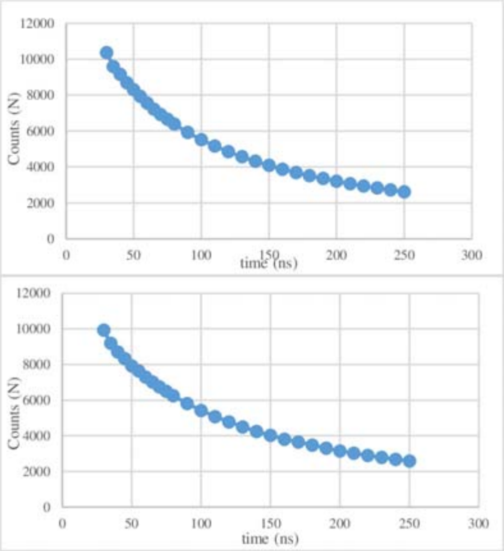

Fig. 24. Time stretching of the T_{Start} and T_{Stop} uncertainty.

The counts/ns relationship was derived based on the maximum possible uncertainty, calculated as 216 / 291.32 ns = 224 counts/ns. This results in a resolution of 4.46 ps per count. For the stop-trigger uncertainty, the counts/ns relationship was similarly derived, yielding 216/252.78 ns = 259.26 counts/ns, yielding a resolution of 3.857 ps per count. The data obtained were conveniently rounded to integer values due to the significant memory requirements for storing floating-point data in a general-purpose MCU. Consequently, the data were stored in two arrays of 28 values each in the MCU's static memory to facilitate processing through software-based linear interpolation. The time was determined using the specific experimental data points (x_1, y_1) and (x_2, y_2), and the relevant elements in the lookup table were identified via linear interpolation, as described by Eq. (8), allowing the corresponding value to be found.

3.4. Complete Implementation of the High-resolution Time-to-digital Converter

As part of the characterization, the minimum input time interval, T_x, that the TDC can measure was analyzed, which is considered the minimum allowable input time. This limitation is due to the initialization time required for processing the interrupt vector in the capture mode of the MCU timers. The interrupt vector handler for the timer of the MSP430F5529 MCU is constrained by a certain number of machine cycles needed by the CPU to execute an interrupt and jump to the appropriate instruction, along with different interrupt modes that introduce latency, quantified in machine cycles as:

- Capture/compare module TA0CCR0 register requires 11 machine cycles;
- Capture/compare module TA0CCR1, TA0CCR2, TA0CCR3, TA0CCR4, TA0CCR5, and TA0CCR6 registers requires 16 machine cycles;
- Timer overflow TA0IFG register flag requires 14 machine cycles.

Therefore, capturing a time interval of 16 machine cycles or fewer, that is, 16 × 250 ns = 4 μs, is insufficient to initialize a time-stretching sequence.

Consequently, this time was considered as initialization time because time intervals shorter than 4 μs cannot detect an interrupt in capture mode, and in any case, the timer's counter overflows. When the measurement of a time interval T_x is initialized, the TA0 timer on physical pin 2.4 is set to count from 0 to 65535 possible counts, with an interrupt in capture mode triggered by the rising edge of the signal. When a time interval occurs, the interrupt vector searches for the correct interrupt vector, which introduces a delay of 11 machine cycles before entering capture mode on the rising edge of the clock. During this process, the exact time is captured upon entering the interrupt and stored in a software variable. Within the same interrupt, another interrupt is configured to operate in descending edge mode, stopping the counter and obtaining the synchronized time value T_s. These code statements consume a total of 4 equivalent code instructions, corresponding to 16 clock cycles, plus an additional 11 clock cycles to access the correct interrupt vector to capture the precise time based on the MCU's resolution when entering capture mode on the descending edge and stopping the counter. Thus, it was concluded that the minimum/maximum time intervals, T_x, that the TDC can capture are presented in Table 3.

Table 3. Time consumption for a stretching sequence.

Time Interval	Time stretching for the start trigger signal	Time stretching for the stop trigger signal
30 ns	0.31096 ms	0.297975 ms
250 ns	0.6545 ms	0.65155 ms

As part of characterization, time stretching was analyzed. The sum of both times provides an overview of the time required for processing the time stretching defined by Eq. (11). Table 3 presents the times for a time interval at its minimum and maximum stretching points.

The maximum uncertainty at both edges never exceeds 1.30605 ms during stretching processing. Therefore, the total processing time of the TDC for measuring a time interval is the time interval plus the stretching processing for the maximum uncertainty, expressed as $T_{\text{total}} = T_x + T_{\text{max\_start-stretch}} + T_{\text{max\_stop-stretch}}$. The characterization of the TDC for measuring a time interval followed the procedure outlined in Section 2.4, using the Keysight 33500B function generator to generate the time interval. An output pulse width of 15 μs was configured, with 5 ns increments between measurements, allowing the synchronized time interval to be measured relative to the MCU clock. The uncertainties generated at both the start and end edges of the time interval T_x were captured and interpreted as a digital value for further analysis.

The TDC implementation was presented as a PCB sub-assembly for the MSP430F5529 MCU development platform. The current consumption from the MCU on its platform in low-power mode (LPM4) was measured at 2.05 μA, while in normal operating mode, it consumed 1.042 mA. The current consumption of the TDC circuit was recorded at 150 μA in idle mode and averaged 1.77 mA during operation.

Finally, the current consumption of the XBee communication module was noted at 38 mA during communication and at 8.95 μA in low-power mode via the sleep mode pin, which was controlled by the physical pin P7.4 on the MCU. In a scenario without using the development board, the proposed TDC and MCU's current operational consumption is estimated to be no more than 3 mA, with an average low consumption of 160 μA.

4. Comparison and Discussion

A direct comparison of the high-resolution TDC with a general-purpose MCU versus related works is summarized in Table 4, highlighting improvements at: 1) achieving practical high-resolution by reducing the quantization errors due to synchronization between the reference clock T_{ref} and the start and stop trigger signals (T_{Start} and T_{Stop}) of the time interval T_x, and 2) a wide measurement range, achieved by the interpolation, time-to-voltage conversion, and time-stretching by measuring separately the time intervals T_s, T_{Start} and T_{Stop} to obtain T_x.

Related works, such as the high-resolution TDC with time-stretching technique with a general-purpose MCU [30], which achieves a practical resolution of 0.17 ns by stretching the time interval by a factor of 1000, resulting in an extensive, time-consuming process for small operating time measurement ranges, On the other hand, A TDC with a vernier technique [33], which achieve a resolution of 2.02 ns with an external oscillator running slightly different to the main MCU oscillator, this proposal ideal, but in a practical application the detection of the coincidence moment between the timers could introduce latency because the comparison algorithm proposed consumes multiple instruction cycles to execute, leading to uncertainty about the exact moment the timers coincide, reducing the precision of any single measurement.

Table 4. Comparison of related TDC MCU-based works.

Work	T_{ref}	Method	Measurement ranges	Resolution	Conversion time
[30]	200 ns	Time-stretching	1 µs to 10 µs	0.17 ns	1 ms to ~10 ms
[33]	200 ns	Vernier	16 µs to 132 µs	2.02 ns	up to ~132 µs
This work	250 ns	Interpolator, time-to-voltage conversion, and time-stretching	10 µs to 16 ms	0.1 - 4.14 ns	650 µs to ~18 ms

The TDC implemented in this work is based on the design of an interpolator, a time-to-voltage conversion stage, and a time-stretching to get high-resolution, the prototype dimensions are 58 mm × 62 mm with the main limitation to measure intervals shorter than 10 µs due to the MCU's embedded digital timer, which requires 38 clock cycles (~9.5 µs) to initialize an interrupt service routine, attributed to the interrupt vector handler of MSP430F5529 MCU embedded timers, these constrains due to certain number of machine cycles needed by the CPU to execute an interrupt service routine and jump to the appropriate instruction as described as follows: a) 11 machine cycles to entering to capture mode on the rising edge of the reference clock for an oncoming time interval T_x, b) another interrupt service routine is configured to operate in descending edge mode, stopping the counter and obtaining the synchronized time value T_s. These code statements consume a total of 4 equivalent code instructions, corresponding to 16 clock cycles, and c) an additional 11 machine cycles to access the correct interrupt vector to capture the precise time based on the MCU's resolution when entering capture mode on the descending edge and stopping the counter.

This can be mitigated by using an MCU with faster interrupt handling or a higher oscillator frequency. The TDC is adaptable to different reference clocks through adjustments in the *RC* network. However, slower clocks significantly increase the time-stretching effect. For example, stretching increases from 650 µs with a 4 MHz reference clock to 17 ms with a 1 MHz reference clock. Thus, applications requiring slow clocks must carefully consider this trade-off.

The uncertainty detected is directly related when measuring start and stop trigger signals (T_{Start} and T_{Stop}), associated with the time stretching technique, pointing out to a maximum start-trigger uncertainty at 216 / 291.32 ns = 224 counts/ns, what it means a 4.46 ps resolution per count, and for the stop-trigger uncertainty, the counts/ns relationship was similarly derived, yielding 216/252.78 ns = 259.26 counts/ns, a resolution of 3.857 ps resolution per count.

The digital values of the start and stop trigger signals are interpolated using a lookup table to linearize them with respect to the stored data, using two known data points. These data points were stored in the MCU's EEPROM, along with the linearized values, to locate new incoming data. This tool helps to address the nonlinear behavior of a capacitor's charging and discharging, enabling precise linearization and correction of the TDC's output response.

As part of the experimental results, it was found that the maximum variability observed for the start-trigger was 2.02 ns over a 250 ns interval. In comparison, the minimum was 0.4469 ns in a 50 ns interval. On the other hand, the maximum variability observed for the stop-trigger was 4.17 ns over a 250 ns time interval. In comparison, the minimum variability recorded was 0.1156 ns over a 30 ns time interval. This shows that the effective resolution of the entire hardware is around 0.1-4.14 ns.

However, another similar approach is the TDCs implemented with an FPGA, which is a more complex solution. Table 5 provides a general overview of recent TDC FPGA-based methods, including the measured ranges and resolutions. It is demonstrated that the FPGA's ability to run at a high clock reference allows high-resolution operation with embedded hardware methods, such as delay-line techniques, pulse-shrinking, or a combination of multiple techniques. The authors do not report power consumption results, but in theory, implementing TDCs in FPGAs often consumes a significant portion of the FPGA fabric due to the use of numerous logic cells and routing resources that would otherwise be used for application logic, thereby increasing power consumption.

A brief discussion of the TDC MCU-based proposal in this work highlights several advantages over FPGA-based implementations. MCUs typically consume less power, making them ideal for battery-operated and portable applications where energy efficiency is crucial. The integration within a general-purpose MCU simplifies system design by reducing the complexity of inter-chip communication and synchronization, which is often a challenge in FPGA-based systems. Furthermore, MCUs generally cost less and have smaller form factors than FPGAs, making them more cost-effective for low- to medium-volume applications. From a development perspective, MCUs provide mature, easy-to-use development environments and debugging tools, facilitating faster prototyping and deployment.

TDCs with MCU-based designs may have limitations in achieving the highest resolution and speed available from FPGA implementations. However, the proposed TDC MCU-based approach offers a balanced solution that combines energy efficiency, integration, ease of development, and cost-effectiveness across a wide range of precision TDC applications, including sensor-to-MCU interfacing.

Table 5. Comparison of TDC FPGA-based works.

Work	T_{ref}	Method	Measurement ranges	Resolution
[24]	1.83 ns	Merged delay lines	50 ns	4.26 ps
[29]	-	Pulse shrinking	11.5 ns	42 ps
[34]	5 ns	Multi-Phase clock TDC	625 ns	625 ps
[35]	1.04 ns	Vernier loop shrinking	10 ns	8.5 ps
[36]	3.3 ns	Multi-Phase clock TDC	5 ns to 205 ns	84 ps

In general, this capability enables the development of lightweight, battery-powered devices that can be deployed across diverse environments, enhancing the practicality and versatility of sensor-based systems in real-world applications.

5. Conclusions

A high-resolution, low-cost TDC was implemented on a general-purpose MCU using analog interpolation and time stretching. This demonstrates the feasibility of achieving high resolution in TDCs with minimal discrete components. In the proposed hardware, the MCU timer measures time intervals synchronized with a 4 MHz reference clock, capturing uncertainties in the start and stop triggers through two analog interpolators. These interpolators use capacitors to store the pulse-width uncertainty in a time-to-voltage conversion, which the MCU processes via time stretching, governed by a current ratio in which the capacitor's discharge current is much lower than its charging current, thereby enhancing resolution. The TDC achieves an effective resolution of 4 ns over an interval range of 10 µs to 16 ms, representing a 60-fold improvement over the MCU's clock resolution. This performance is comparable to a 240 MHz clock.

This proposal is particularly well-suited for sensor-to-MCU interfacing applications, where portable solutions are essential. The combination of high resolution and low-cost implementation makes it ideal for scenarios that require efficient data acquisition from various sensors while minimizing the footprint and power consumption.

References

[1]. K. Park, J. Park, 20 ps resolution time-to-digital converter for digital storage oscilloscopes, in *Proceedings of the IEEE Nuclear Science Symposium Conference Record*, 1998, pp. 876-881.

[2]. P. Palojarvi, K. Maatta, J. Kostamovaara, Integrated time-of-flight laser radar, *IEEE Transactions on Instrumentation and Measurement*, Vol. 46, Issue 4, 1997, pp. 996-999.

[3]. M. Maamoun, R. Beguenane, A. Benbelkacem, A 3ps resolution time-to-digital converter in low-cost FPGA for laser rangefinder, in *Proceedings of the World Congress on Engineering*, Vol. 1, 2017, pp. 259-263.

[4]. K. Yoshioka, et al., A 20-ch TDC/ADC hybrid architecture LiDAR SoC for 240×96 pixel 200-m range imaging with smart accumulation technique and residue quantizing SAR ADC, *IEEE Journal of Solid-State Circuits*, Vol. 53, Issue 11, 2018, pp. 3026-3038.

[5]. A. E. Stevens, R. P. Van Berg, J. Van der Spiegel, H. Williams, A time-to-voltage converter and analog memory for colliding beam detectors, *IEEE Journal of Solid-State Circuits*, Vol. 24, Issue 6, 1989, pp. 1748-1752.

[6]. M. P. Mattad, H. Guhilot, R. K. Kamat, Area efficient time to digital converter (TDC) architecture with double ring-oscillator technique on FPGA for fluorescence measurement application, in *Proceedings of the IEEE Recent Advances in Intelligent Computational Systems Conference (RAICS'11)*, 2011, pp. 260-263.

[7]. K. Naresh, Applications of fluorescence spectroscopy, *Journal of Chemical and Pharmaceutical Sciences*, Special Issue 5, 2014, pp. 18-21.

[8]. I. Nissinen, J. Nissinen, J. Holma, J. Kostamovaara, A TDC-based 4×128 CMOS SPAD array for time-gated Raman spectroscopy, in *Proceedings of the European Solid-State Circuits Conference (ESSCIRC'14)*, 2014, pp. 139-142.

[9]. P. Vines, et al., High-performance planar germanium-on-silicon single-photon avalanche diode detectors, *Nature Communications*, Vol. 10, Issue 1, 2019, 1086.

[10]. C.-C. Chen, W.-F. Lu, C.-C. Tsai, P. Chen, A time-to-digital-converter-based CMOS smart temperature sensor, in *Proceedings of the IEEE International Symposium on Circuits and Systems*, 2005, pp. 560-563.

[11]. C. -C. Chen, P. Chen, An accurate CMOS delay-line-based smart temperature sensor for low-power low-cost systems, *Measurement Science and Technology*, Vol. 17, Issue 4, 2006, pp. 840-846.

[12]. M. P. Mattada, S. M. Magadum, H. Guhilot, Identification of hotspots on FPGA using time-to-digital converter and distributed tiny sensors, in *Proceedings of the International Symposium on Physics and Technology of Sensors (ISPTS'15)*, 2015, pp. 235-239.

[13]. S. Henzler, Time-to-Digital Converters, *Springer*, 2010.

[14]. M. P. Mattada, H. Guhilot, Time-to-digital converters – a comprehensive review, *International Journal of Circuit Theory and Applications*, Vol. 49, Issue 3, 2021, pp. 778-800.

[15]. S. Tancock, E. Arabul, N. Dahnoun, A review of new time-to-digital conversion techniques, *IEEE Transactions on Instrumentation and Measurement*, Vol. 68, Issue 10, 2019, pp. 3406-3417.

[16]. D. I. Porat, Review of sub-nanosecond time-interval measurements, *IEEE Transactions on Nuclear Science*, Vol. 20, Issue 5, 1973, pp. 36-51.

[17]. Z. Xu, M. Miyahara, A. Matsuzawa, A 1 ps-resolution integrator-based time-to-digital converter using a SAR-ADC in 90nm CMOS, in *Proceedings of the IEEE International New Circuits and Systems Conference (NEWCAS'13)*, 2013, pp. 1-4.

[18]. M. Kim, H. Lee, J. -K. Woo, N. Xing, M. -O. Kim, S. Kim, A low-cost and low-power time-to-digital converter using triple-slope time stretching, *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 58, Issue 3, 2011, pp. 169-173.

[19]. P. Chen, C. -C. Chen, Y. -S. Shen, A low-cost low-power CMOS time-to-digital converter based on pulse stretching, *IEEE Transactions on Nuclear Science*, Vol. 53, Issue 4, 2006, pp. 2215-2220.

[20]. M. Rezvanyvardom, F. E. Farshidi, A novel cyclic time-to-digital converter based on triple-slope interpolation and time amplification, *Radioengineering Journal*, Vol. 24, Issue 3, 2015, pp. 800-807.

[21]. E. Raisanen-Ruotsalainen, T. Rahkonen, J. Kostamovaara, A time digitizer with interpolation based on time-to-voltage conversion, in *Proceedings of the 40th Midwest Symposium on Circuits and Systems*, 1997, pp. 197-200.

[22]. J. Kalisz, M. Pawlowski, R. Pelka, A multiple-interpolation method for fast and precise time digitizing, *IEEE Transactions on Instrumentation and Measurement*, Vol. 35, Issue 2, 1986, pp. 163-169.

[23]. Y. -H. Chen, A high-resolution FPGA-based merged delay line TDC with nonlinearity calibration, in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS'13)*, 2013, pp. 2432-2435.

[24]. Y. Wang, Q. Cao, C. Liu, A multi-chain merged tapped delay line for high-precision time-to-digital converters in FPGAs, *IEEE Transactions on Circuits and Systems II: Express Briefs*, Vol. 65, Issue 1, 2018, pp. 96-100.

[25]. P. Dudek, S. Szczepanski, J. V. Hatfield, A high-resolution CMOS time-to-digital converter utilizing a Vernier delay line, *IEEE Journal of Solid-State Circuits*, Vol. 35, Issue 2, 2000, pp. 240-247.

[26]. N. Xing, H. Song, D. -K. Jeong, S. Kim, A PVT-insensitive time-to-digital converter using fractional difference Vernier delay lines, in *Proceedings of the IEEE International SOC Conference (SOCC'09)*, 2009, pp. 43-46.

[27]. L. Vercesi, A. Liscidini, R. Castello, Two-dimensions Vernier time-to-digital converter, *IEEE Journal of Solid-State Circuits*, Vol. 45, Issue 8, 2010, pp. 1504-1512.

[28]. C. Chen, S. Meng, Z. Xia, G. Fang, H. Yin, Pulse shrinking time-to-digital converter for UWB application, *Journal of Electronics (China)*, Vol. 31, Issue 3, 2014, pp. 180-186.

[29]. R. Szplet, K. Klepacki, An FPGA-integrated time-to-digital converter based on two-stage pulse shrinking, *IEEE Transactions on Instrumentation and Measurement*, Vol. 59, Issue 6, 2010, pp. 1663-1670.

[30]. L. Bengtsson, Implementation of high-resolution time-to-digital converter in 8-bit microcontrollers, *Review of Scientific Instruments*, Vol. 83, Issue 4, 2012, 045107.

[31]. D. Cox, Implementing ohmmeter/temperature sensor, Application Note AN512, *Microchip Technology Inc.*, 1997.

[32]. B. Merritt, MSP430-based digital thermometer, Application Report SLAA038, *Texas Instruments*, 1999.

[33]. L. Bengtsson, Implementation of Vernier TDCs in 8-bit microcontrollers, in *Proceedings of the IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2014, pp. 1-4.

[34]. Y. Wang, X. Cheng, D. Li, W. Zhu, C. Liu, A linear time-over-threshold digitizing scheme and its 64-channel DAQ prototype design on FPGA for a continuous crystal PET detector, *IEEE Transactions on Nuclear Science*, Vol. 61, Issue 1, 2014, pp. 99-106.

[35]. J. Zhang, D. Zhou, An 8.5-ps two-stage Vernier delay-line loop shrinking time-to-digital converter in 130-nm flash FPGA, *IEEE Transactions on Instrumentation and Measurement*, Vol. 67, Issue 2, 2018, pp. 406-414.

[36]. P. Chen, H. C. Cheng, A. Widodo, W. X. Tsai, A PVT insensitive field programmable gate array time-to-digital converter, in *Proceedings of the IEEE Nordic-Mediterranean Workshop on Time-to-Digital Converters (NoMe TDC'13)*, 2013, pp. 1-4.