On Tracking Time with Better Resolution and Range in Batteryless Systems

Eren Yildiz
Ege University
Izmir, Turkiye
eren.yildiz@ege.edu.tr

Davide Cavedon
University of Trento
Trento, Italy
davide.cavedon@studenti.unitn.it

Kasim Sinan Yildirim
University of Trento
Trento, Italy
kasimsinan.yildirim@unitn.it

ABSTRACT

Accurate and failure-resilient timekeeping is crucial for time-sensitive sensing operations and task scheduling in batteryless systems. The main challenge lies in measuring the duration of power failures precisely. Remanence timekeepers offer a simple and energy-efficient solution by exploiting the discharging characteristics of capacitors to measure the power failure duration. However, existing remanence timekeeping designs in the literature can measure either a shorter power failure duration with a higher resolution or a longer power failure duration with a lower resolution. In this paper, we focus on this trade-off and study how to design an ideal timekeeper offering high measurement resolution and range for batteryless systems. We consider CHRT (cascaded hierarchical remanence timekeeper), the de facto remanence timekeeper for batteryless systems, and improve its design to increase its measurement resolution during longer power failure durations. We evaluate the benefits and trade-offs of our new design. Finally, based on our evaluations, we outline potential improvements and research directions for timekeeping in batteryless systems.

CCS CONCEPTS

• **Computer systems organization** → **Embedded systems**; • **Hardware** → **PCB design and layout**.

KEYWORDS

Energy Harvesting, Batteryless Sensors, Timekeeping

ACM Reference Format:
Eren Yildiz, Davide Cavedon, and Kasim Sinan Yildirim. 2024. On Tracking Time with Better Resolution and Range in Batteryless Systems. In *International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems (ENSsys '24), November 4–7, 2024, Hangzhou, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3698384.3699617

1 INTRODUCTION

A batteryless device can operate while its capacitor's voltage exceeds the operating threshold. When it consumes the stored energy in the capacitor, a power failure occurs which turns off the device. During the off period, the device can only charge its capacitor to turn on again. Each power failure resets the volatile states of the processor and the peripherals. Specifically, the contents of the timer subsystem registers are lost. This situation causes the device to lose its *time notion* since it cannot measure its off time (i.e., charging time) when it is off and charging. However, robust timekeeping is crucial since batteryless devices need to keep track of time for time-sensitive sensing operations [10, 14] and task scheduling [9, 11]. For instance, sensed data expires when it cannot be consumed within a specific time frame due to lengthy charging delays [12, 13]. Another example is the event-driven periodic computations [11], whose deadlines should be considered for satisfying application demands and ensuring systems energy efficiency.

Remanenance timekeepers are de facto solutions for measuring off time in batteryless systems. A remanence timekeeper uses an RC circuit to measure the duration of power failures by utilizing capacitor discharging characteristics over a resistor. For instance, CUSTARD [8] estimates power failure duration by monitoring the timekeeping capacitor voltage level after power failure. By utilizing the relationship between the voltage drop and time, CUSTARD can predict the device's off-time with millisecond precision. However, as the power failure duration increases, the voltage drop in the capacitors slows down. This results in a significant decrease in timekeeping range and accuracy. CHRT [5] addressed this issue by employing an array of different-sized capacitors arranged hierarchically. It can measure the duration of longer power failures by dynamically switching between different RC circuits, to increase timekeeping range.

As new applications become feasible to run without batteries, we will require a higher timekeeping range and resolution. For instance, networked applications [1, 4, 6] require time synchronization at least in the order of milliseconds to duty cycle their activities and communicate efficiently. However, maintaining millisecond precision and increasing timekeeping range simultaneously introduces significant overheads in existing CHRT design. As we outline in the next sections, increasing the measurement precision of CHRT introduces substantial hardware area overhead, requires a large number of GPIO pins to be connected to the microcontroller (MCU), adds software complexity, and energy overheads due to analog-to-digital (ADC) conversion.

In this paper, we look for an ideal timekeeper offering high measurement resolution and range for batteryless systems. We study the trade-off between timekeeping range and precision. We review the existing CHRT design and improve its resolution during longer power failure durations. We evaluate our modifications and outline the drawbacks and limitations of our modified design. Finally, based on our evaluations, we outline potential improvements and research directions for better timekeeping in batteryless systems.

ENSsys '24, November 4–7, 2024, Hangzhou, China

ACM ISBN 979-8-4007-1296-8/24/11.
https://doi.org/10.1145/3698384.3699617

2 ARE TIMEKEEPERS GOOD ENOUGH?

Several prior works addressed the challenge of measuring the duration of power failures in batteryless systems. Some studies explored approaches that do not require dedicated hardware but use specific characteristics of widely available hardware blocks in any embedded system. For instance, TARDIS [8] exploits SRAM remanence and relates power failure durations with the decay rate of data stored in SRAM. Alsubhi et al. [2] presented a solution that estimates power failure duration by monitoring the stabilization level of the device's oscillator. However, these solutions can measure only very short power failure durations and offer low accuracy and resolution. Remanence timekeepers [3, 5, 7, 8] use dedicated RC circuits to track power failure duration, providing improved accuracy and resolution compared to the mentioned solutions.

In the following sections, we consider the de facto remanence timekeeper CHRT [5], employing a hierarchical capacitor array to offer a longer timekeeping range with different resolution levels. CHRT places the timekeeping capacitors tier by tier, with the capacitor size increasing at each tier. The lowest tier has the smallest capacitor, while the highest tier has the largest capacitor. CHRT monitors the voltage level of each tier capacitor using a voltage comparator. This comparator controls the discharge switch of the upper tier to ensure that the capacitor of the next tier is discharged when the capacitor of the relevant tier is discharged. When the capacitor of the lower tier is discharged, the discharge switch of the upper tier is closed and the capacitor in the upper tier discharges through a resistor. This process continues until all capacitors in the tiers are discharged. Additionally, each tier is connected to the MCU via an analog GPIO PIN so that the MCU can monitor the voltage level of the capacitors in all tiers. Moreover, there is a switch for recharging the discharged capacitors after each power failure, and a GPIO connection allows this switch to be controlled by the MCU. Therefore, CHRT takes ADC measurements starting from the lowest tier after a power failure to determine the last discharged tier. After estimating the time according to the measured ADC values, the MCU closes the charge switches of the relevant tiers to recharge the capacitors of the discharged tiers, ensuring that the system is ready for the next power failure.

Despite its simple structure and energy-efficient operation, CHRT has several deficiencies. We explain them as follows.

(1) Too Many GPIO Pin Connections. CHRT reserves multiple GPIO pins of the MCU to track how many capacitors are depleted after power failure and reconfigure the tiers of the timekeeping circuit. For instance, the 4-tier version of CHRT requires 16 GPIO pins connection. The number of dedicated pins increases as the number of capacitors and tiers increases.

(1) Potential Improvement: Reduce the number of dedicated GPIO pins for timekeeping.

(2) Software and ADC Overheads. CHRT uses GPIO pins for two things. Firstly, it performs ADC measurements to obtain the current voltage level of the capacitors. It starts from the lowest tier. If that tier's capacitor is depleted, the system moves to the next tier until it finds one with a partially charged capacitor. The cost of this operation increases with the number of depleted tiers. Secondly, it uses GPIO pins to control the switches connected to the RC circuits. It uses these pins to charge the capacitors one by one, resulting in increased complexity.

Table 1: Time range and resolution according to the capacitor size and tiers.

Cap.	Time Range (sec)					Resolution (ms)
	Single	5-Tier	10-Tier	25-Tier	100-Tier	
2.2nF	0.010934	0.05467	0.10934	0.27335	1.0934	0.0128127
22nF	0.10934	0.5467	1.0934	2.7335	10.934	0.128127
220nF	1.0934	5.467	10.934	27.335	109.34	1.28127
2.2μF	10.934	54.67	109.34	273.35	1093.4	12.8127
10μF	49.968	546.7	499.68	1249.2	4996.8	58.2399

Table 2: Approximate PCB sizes for designs with a different number of tiers.

	5-Tier	10-Tier	25-Tier	100-Tier
Size (mm x mm)	15x30	30x30	45x55	100x100

(2) Potential Improvement: Reduce software complexity by automatically charging timekeeping capacitors without the need for extra switches. Reduce the energy overhead by removing unnecessary ADC measurements for the depleted tiers.

(3) Gradually Degraded Resolution. CHRT extends the measurable time range by using progressively larger capacitors at higher tiers, resulting in reduced measurement resolution at these tiers. For instance, while the measurement resolution is in milliseconds at the first stage, it might be in seconds or minutes at the last stage. However, several applications (e.g., network time synchronization protocols [4]) require a fixed precision timekeeping.

(3) Potential Improvement: Keep the measurement resolution fixed at each tier to ensure precise measurements even with longer power failure durations.

3 OPTIMIZING CHRT DESIGN FOR BETTER TIMEKEEPING RESOLUTION

Considering the potential improvements mentioned in the previous section, we modified the CHRT design and created CHRT++ which provides a fixed timekeeping resolution during longer power failure durations. Before presenting CHRT++, we explore the design of CHRT in more detail to highlight the impact of different design choices on timekeeping.

3.1 Exploring the CHRT Design

Some design choices, such as timekeeping capacitor size or number, are crucial for optimal timekeeper design. While a small timekeeping capacitor offers high precision, it has a short timekeeping range. Similarly, using a big capacitor offers a longer time range but with low precision. CHRT overcomes a part of this issue using a cascaded hierarchical capacitor design. This design has the smallest capacitor at the first tier for highly precise measurement during short power failures. The capacitor sizes are increased tier by tier to measure longer power failures. However, this brings a gradually reduced resolution with the higher tiers (or longer power failures).

Table 1 presents the time range and resolution performance of different capacitors when discharging from 3V to 0.25V through a

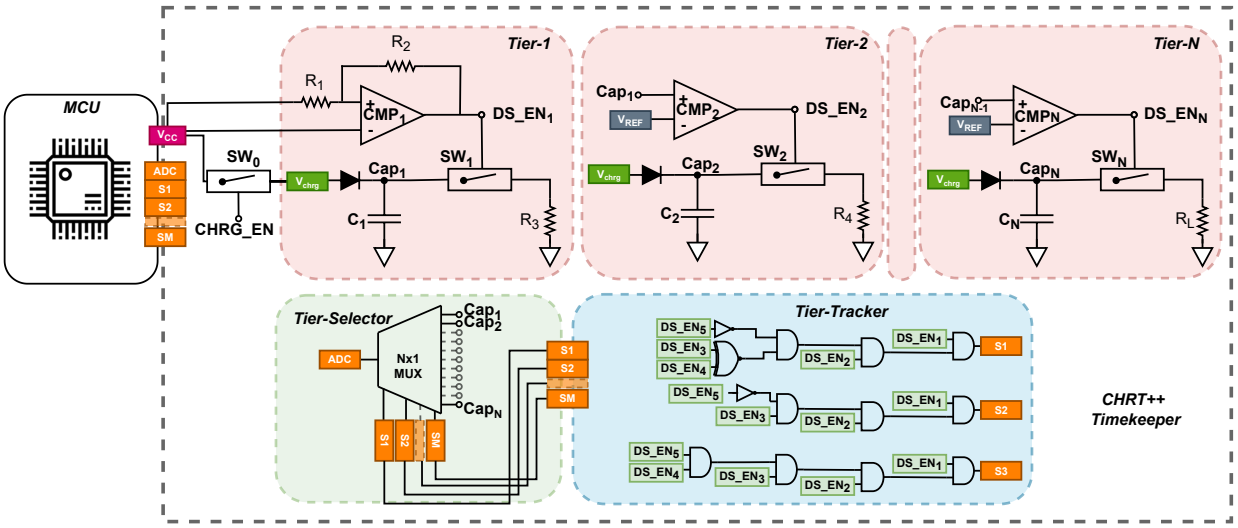

Figure 1: CHRT++ timekeeping architecture.

Table 3: The number of tiers and GPIO pins.

	5-Tier	10-Tier	25-Tier	100-Tier
GPIO PINs	19	34	79	304

2MΩ resistor. This table clearly shows the trade-off between time range and resolution. For instance, a 2.2μF capacitor offers the best resolution at 12μs but can only measure power failures up to 10 ms. On the other hand, a 10μF capacitor can measure power failures lasting almost 50 seconds but with a coarser 58 ms resolution.

3.1.1 How can we ensure a long range with high resolution? When we consider the 5-tier CHRT design with the capacitors specified in Table 1, it can ensure an approximate time range of 62 seconds. However, after the 3rd stage (approximately 1.2 seconds), it will fail to meet the resolution requirement. One possible solution is to use the same capacitor in each stage, but this significantly reduces the measurement range to 5.467 seconds. To increase the range while maintaining resolution, the number of tiers can be increased. Table 1 shows the time range for 5, 10, 25, and 100-tier versions of CHRT using the same capacitor for each tier, demonstrating that increasing the number of tiers also increases the time range. However, increasing the number of tiers leads to three main problems for CHRT.

(1) *Increased Hardware Size.* We have summarized the approximate PCB sizes of various CHRT designs with different numbers of tiers in Table 2. For instance, 25-tier CHRT has a PCB size of approximately 50x50mm. If we increase the number of tiers to 100, the size would be approximately 100x100mm.
(2) *ADC Cost.* When tier size increases, another issue that arises is the energy overhead due to ADC measurement. In the current version of CHRT, after a power failure, it will take ADC measurement for each tier, which means one measurement for each of the 25 tiers. This would significantly increase the energy overhead of the timekeeping.
(3) *Required GPIO PINs.* Table 3 shows the number of GPIO pins based on the tier size for CHRT. CHRT requires dedicated GPIO pins for functions such as enabling/disabling, charging, ADC measurements, and other switch controls for each tier. As a result, the pin requirement significantly increases with tier size. For example, CHRT requires 34, 79, and 304 GPIO pins for 10, 25, and 100 tier versions, respectively. Considering that the MSP430fr5994, a commonly used microcontroller for batteryless systems, has a total of 64 GPIO pins, it appears that the 25 and 100-tier versions cannot be controlled even with a single dedicated microcontroller.

In short, increasing the number of tiers can meet the range and resolution requirements for CHRT. However, practical limitations make this option unfeasible due to the issues we mentioned. Therefore, we require significant improvements to the current version of CHRT to address these challenges.

3.2 CHRT++ Design

We designed CHRT++ by using the same size capacitors in each tier to preserve the required resolution for the application. CHRT++ consists of three main parts as shown in Figure 1: capacitor tiers, tier-tracking logic, and tier-selector.

CHRT++ Tiers. We designed our system's tier architecture on top of CHRT with notable differences that reduce the required GPIO pins. In our design, we used only one switch and one dedicated diode for each tier. This allows us to charge all tier capacitors simultaneously using only one GPIO pin (*CHRG_EN*). The diodes prevent reverse currents between capacitors, so each tier capacitor starts discharging right after its $DS\_EN_{\#}$ signal is triggered. The $DS\_EN_{\#}$ signal represents whether the previous tier's capacitor is depleted or not. In each tier, one comparator tracks the voltage level of the previous tier's capacitor. When the voltage level of the previous tier's capacitor drops below the V_{REF}, the comparator triggers the next tier via $DS\_EN_{\#}$, connecting the discharging signal of each tier to the previous tier. Only the first tier's comparator

tracks the VCC port of the MCU to detect a power failure. We also utilized the hysteresis feature of the comparator to prevent voltage fluctuation on VCC from affecting the system.

CHRT++ Tier-Tracker. The tier-tracker is an essential part of CHRT++ as it provides information on the most recently discharged tier to the tier-selector and the MCU through tier tracker bits (S1 to SM in Figure 1). This layer uses a logic circuit and takes discharge signals ($DS\_EN_{\#}$) as input to determine the latest discharged tier. This allows CHRT++ to avoid the direct connection of capacitors from each tier to the MCU. Figure 1 shows a Tier-tracker logic circuit for the 5-tier version. It's important to note that as the number of tiers increases, the complexity of the Tier-tracker and the number of logic gates used also increases. While this reduces the GPIO pin requirement, it also raises complexity and cost. However, considering that the 25 and 100-tier versions of CHRT cannot be controlled by a microcontroller due to the GPIO requirement, the Tier-tracker in CHRT++ offers a more practical solution.

CHRT++ Tier-Selector. The tier-selector is responsible for connecting the latest tier's capacitor to the ADC pin of the MCU to measure the voltage drop. Our system utilizes a multiplexer to achieve this. The tier-tracker bits act as selective bits of the multiplexer. Furthermore, these bits are directly connected to the MCU to provide information about the tier level. As a result, our system ensures that the MCU receives the latest tier-level information and can measure the correct capacitor, eliminating the need to check each capacitor if it is empty via multiple ADCs. The size of the multiplexer and the number of tier tracker bits depend on the tier number in the design.

Calculating Power Failure Duration. Remanence timekeepers estimate the duration of a power failure by using the capacitor discharge formula given in Equation 1. In this equation, the discharging resistor (R), capacitance (C), and initial voltage (V_0) are known and constant variables. The MCU can calculate the discharging time of the capacitor by reading the current voltage level of the capacitor (V_{curr}) via an ADC. Since the capacitors in each tier are equal in the system's design, the depletion time of a capacitor can be calculated by substituting V_l for V_{curr} in Equation 2. Here, V_l represents the lower threshold that triggers the next tier. Consequently, the system calculates the power failure duration by using tier-tracker bits (S_0 to S_M) and Equations 1 and 2, as shown in Equation 3.

$$t_{dchrg} = RCln(\frac{V_0}{V_{curr}}) \tag{1}$$

$$t_{tier} = RCln(\frac{V_0}{V_l}) \tag{2}$$

$$t_{pf} = (S_M S_{M-1}..S_0)_{dec} * t_{tier} + RCln(\frac{V_0}{V_{curr}}) \tag{3}$$

3.3 CHRT++ Hardware Prototype

We designed a 4-tier version of CHRT++ using 220nF capacitors for each tier. We considered Commercial Off-The-Shelf (COTS) components for our PCB design presented in Figure 2. We utilized one Texas Instruments nano power voltage comparator for each tier, totaling four. We chose this particular voltage comparator for its 75 nA quiescent current. Furthermore, we used five analog switches with less than 0.01μW power consumption - one for charging all the capacitors and four for discharging the capacitors in each tier. Each tier also contains a BAT43W-TP diode with a 260mV forward voltage and a 2MΩ resistor for discharging. For the Tier-tracker, we employed one 74AUP1G86GW XOR gate and four SN74LVC08ARGYR AND gates to generate tier-tracker bits for the 4-tier implementation. Finally, we used a TMUX1104DQAR 4x1 multiplexer with a 3pA leakage current.

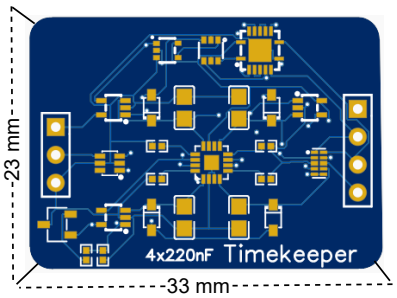

Figure 2: 4-tier CHRT++ prototype design

4 EVALUATION AND RESULTS

We evaluated CHRT++ in two parts: self-evaluation and a comparison with CHRT. During our evaluations, we utilized a 4-tier version of CHRT++, the prototype design of which is shown in Figure 2. In the self-evaluation, we used LTSpice simulations to test the charging, discharging, and tier-tracking behaviors of CHRT++. We could not perform real hardware experiments since our PCB was under production at the time of submission. In the second part, we compared CHRT++ with CHRT by considering energy consumption, ADC measurements, and GPIO pin requirements. We based our power consumption numbers on the component datasheets when comparing with CHRT. To calculate the ADC energy cost, we measured the ADC energy cost of the MSP430FR5994 using the Energy Trace tool of Code Composer Studio.

4.1 Self-evaluation of CHRT++

In our analysis of CHRT++'s performance, we examined how the system behaved during different durations of power failure. We conducted simulations and focused on the behavior of the system when power was interrupted. To simulate this, we used an adjustable voltage source to replicate the on-and-off conditions of the MCU. The results of our simulations are shown in Figure 3. The top graph in the figure illustrates the output voltage of the MCU, referred to as VCC, which charges the timekeeper. When the VCC voltage drops to 0V, it indicates a power failure, triggering the activation of CHRT++. The second graph displays the $DS\_EN_{\#}$ signals of each tier, which track the previous tier capacitor to initiate the current tier. The third graph represents the status of the tier-tracker bits ($S_{\#}$) that inform the MCU about the latest triggered tier. Lastly, the last graph shows the output of CHRT++ needed for the ADC measurement taken by the MCU.

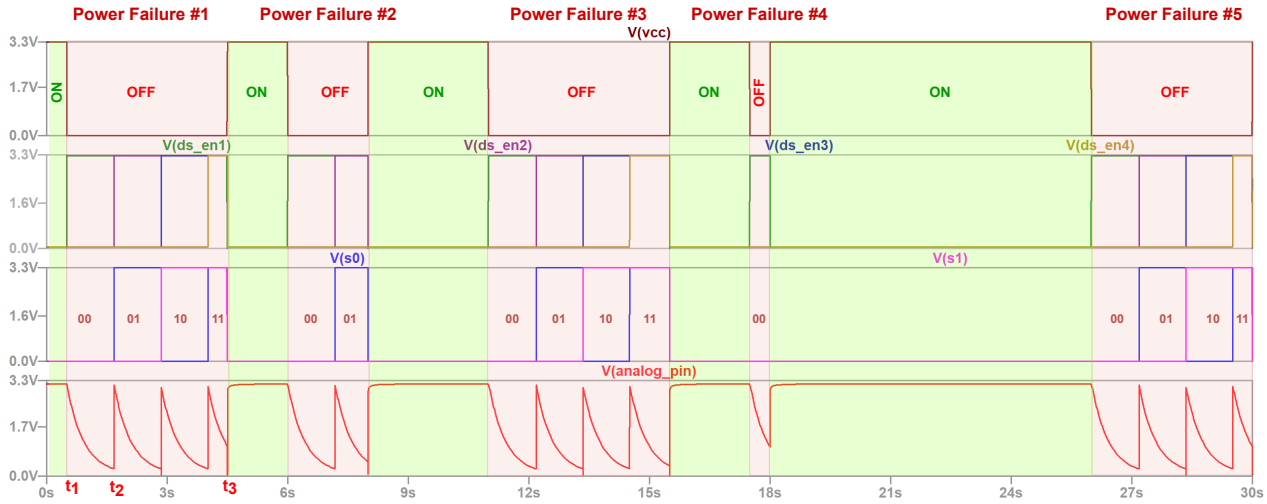

Figure 3: LTSpice simulation of CHRT++ with various power failure durations.

Table 4: Energy Cost of CHRT++ and CHRT. (C_{tier}: 220nF)

	4-tier		25-tier		100-tier	
	CHRT	CHRT++	CHRT	CHRT++	CHRT	CHRT++
Energy Cons. (μJ)	108.01	49.49	4085.50	1799.58	65071.91	28497.28
Num. of ADC	4	1	25	1	100	1

In Figure 3, the first power failure occurs at time t_1. At this point, CHRT++ is triggered by the *ds_en*1 signal, and the first tier capacitor starts discharging. The Tier-tracker of CHRT++ sets the S_0 and S_1 bits as 00, indicating that none of CHRT++'s tiers have depleted capacitors. At time t_2, the first tier capacitor is depleted, prompting CHRT++ to trigger the second tier via the *ds_en2* signal, and the second tier capacitor begins discharging. The Tier-tracker sets the tier-tracker bits as 01 to indicate that the first-tier capacitor is depleted. CHRT++ continues to discharge capacitors tier by tier until the MCU wakes up. When the device wakes up at time t_3, it reads the tier-tracker bits through its GPIO pins. With the tier-tracker bits set as 11, the device understands that three tiers' capacitors are depleted, and the ADC result belongs to the fourth tier capacitor. By utilizing the known total discharging time of a single capacitor, the MCU easily calculates the total power failure time using Equation 3. In a similar manner, CHRT++ tracks the total power failure duration by observing S_0, S_1, and the *analog_pin* for other power failures. It's important to note that the total power failure duration of the third power failure represents the maximum range of the current CHRT++ implementation. To increase it, we would need to increase the tier numbers.

4.2 Comparison with CHRT

We compared CHRT++'s energy consumption, required pins, and ADC measurement needs with those of CHRT. We specifically looked at the CHRT version that uses the same capacitor in each tier to maintain measurement resolution. Therefore, we did not include the time range in our evaluation since it would be the same for both approaches. To calculate the metrics, we used the energy consumption data from the datasheets of each component for various tier versions. For the 25 and 100-tier versions, we calculated the approximate energy consumptions since the complexity of the tier trackers of both CHRT and CHRT++ is significantly increased. Table 4 presents the calculated energy consumption and required ADC measurement numbers. CHRT++ reduces power consumption by over 50% for all layer configurations. This is achieved by decreasing the number of GPIO pins that control the hardware through the MCU and reducing the number of logic gates and switches connected to these pins. In addition, CHRT's ADC requirement varies depending on the depleted tiers, as it measures each tier's capacitor until it finds the undepleted one. However, CHRT++ estimates the time with one ADC for any depleted tier. Figure 4 depicts the required GPIO pins for CHRT++ and CHRT. With an increase in tier numbers, CHRT's pin requirement escalates significantly, and beyond 20 tiers, CHRT is unable to be controlled by the MSP430FR5994, which is a common microcontroller for batteryless applications. Conversely, CHRT++ reduces the pins requirement by up to 30 times and can monitor 100-tier versions using only 9 pins.

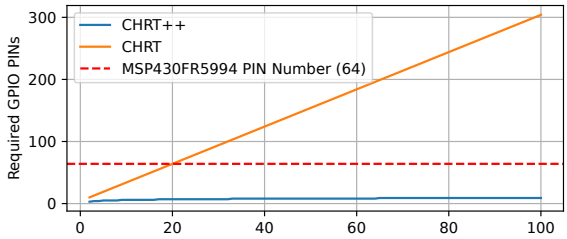

Figure 4: Number of required GPIO pins in CHRT and CHRT++ according to the tier size.

5 DISCUSION

Software-based Timekeeping. Some software-based timekeeping methods track time without needing specific hardware like SQUID, TARDIS [8], and oscillator-based timekeepers [2]. However, most

of these solutions have a very short time range, typically just a few seconds. Only SQUID [15] can offer a long time range with high resolution by tracking incoming power levels when the device is on. However, it depends on stable ambient energy to provide accurate estimates, which makes it unreliable under unstable energy conditions. Therefore, while this approach shows promise in terms of resolution, range, and simplicity, it still needs further development to ensure reliable accuracy.

Low Power Timers. Another alternative to using remanence timekeepers for time tracking in batteryless systems is the ultra-low power timer, known as the TPL5110. This timer consumes an ultra-low current of 35nA and generates pulses with an adjustable resolution ranging from 100ms to 7200s. However, these timers require a counter to be used. Additionally, it's worth noting that the maximum resolution provided by the TPL5110 is significantly lower than what CHRT++ can offer.

Optimizing Main Capacitor of the System. In order to balance the resolution and the number of tiers, we can play with the main energy storage capacitor size. Instead of using too many cascade capacitors to maintain resolution, we can reduce the size of the main capacitor. This allows the system to charge quickly, enabling the timekeeper to measure off time with high resolution without requiring high tiers. However, this increases the software design burden for programmers, as they will need to use smaller tasks or frequent checkpoints to prevent non-termination.

Limitations of CHRT++. Despite our improvements in CHRT design, CHRT++ is still far from an ideal timekeeper due to the extremely large tier levels and the complexity of the tier-tracker circuit. While CHRT++ can provide high resolution with its simple design and fewer GPIO pin requirements for short power failures, it suffers from circuit complexity and cost that increase with the number of tiers for longer power failures.

6 CONCLUSION AND FEATURE WORK

In this paper, we explored the trade-offs in designing an ideal timekeeper for batteryless systems to provide both high measurement resolution and extended range. By examining and enhancing the CHRT [5] (Cascaded Hierarchical Remanence Timekeeper), a widely used timekeeping solution for batteryless environments, we have developed a new design that improves measurement resolution during extended periods of power failure.

Our enhanced design not only increases measurement precision but also significantly reduces GPIO pin requirements by up to 30 times. This reduction enables the development of larger-tier timekeepers, which can support more complex and advanced applications in batteryless systems. However, despite these improvements, the increased circuit complexity that accompanies the design highlights the challenges of achieving both high resolution and long-range timekeeping.

These findings stress the difficulties in creating a timekeeper that balances precision, range, and simplicity. As a result, further effort is needed to address these challenges and advance the development of efficient, durable timekeeping solutions for energy-constrained, intermittent computing environments. This includes optimizing the design to manage complexity without compromising performance and exploring new approaches that could simplify the architecture while maintaining high accuracy and reliability.

In our future work, we will concentrate on simplifying the hardware of CHRT++ for higher tiers and improving our prototype design. Furthermore, we will provide a more comprehensive evaluation of CHRT++ in a real energy harvesting environment.

ACKNOWLEDGEMENT

This work was partially supported by the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART").

REFERENCES

[1] Saad Ahmed, Bashima Islam, Kasim Sinan Yildirim, Marco Zimmerling, Przemysław Pawełczak, Muhammad Hamad Alizai, Brandon Lucia, Luca Mottola, Jacob Sorber, and Josiah Hester. The internet of batteryless things. *Communications of the ACM*, 67(3):64–73, 2024.

[2] Arwa Alsubhi, Nicole Tobias, Simeon Babatunde, and Jacob Sorber. Can crystal oscillators keep time without power? In *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, pages 84–85, 2020.

[3] Alberto Rodriguez Arreola, Theodoros D Verykios, Marco A Gurrola Navarro, and Carlos F Calvillo Cortes. Federated time persistency in intermittently powered iot systems. *Journal of Systems Architecture*, 130:102667, 2022.

[4] Eren Çürük, Kasim Sinan Yıldırım, Przemyslaw Pawelczak, and Josiah Hester. On the accuracy of network synchronization using persistent hourglass clocks. In *Proceedings of the 7th International Workshop on Energy Harvesting & Energy-Neutral Sensing Systems*, pages 35–41, 2019.

[5] Jasper de Winkel, Carlo Delle Donne, Kasim Sinan Yildirim, Przemysław Pawełczak, and Josiah Hester. Reliable timekeeping for intermittent computing. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 53–67, 2020.

[6] Jasper De Winkel, Haozhe Tang, and Przemysław Pawełczak. Intermittently-powered bluetooth that works. In *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services*, pages 287–301, 2022.

[7] Vishal Deep, Vishak Narayanan, Mathew Wymore, Daji Qiao, and Henry Duwe. Harc: A heterogeneous array of redundant persistent clocks for batteryless, intermittently-powered systems. In *2020 IEEE Real-Time Systems Symposium (RTSS)*, pages 270–282. IEEE, 2020.

[8] Josiah Hester, Nicole Tobias, Amir Rahmati, Lanny Sitanayah, Daniel Holcomb, Kevin Fu, Wayne P. Burleson, and Jacob Sorber. Persistent clocks for batteryless sensing devices. *ACM Transactions onn Embeddded Computing Systems*, 15(4):77:1–77:28, August 2016.

[9] Bashima Islam and Shahriar Nirjon. Scheduling computational and energy harvesting tasks in deadline-aware intermittent systems. In *2020 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 95–109. IEEE, 2020.

[10] Vito Kortbeek, Kasim Sinan Yildirim, Abu Bakar, Jacob Sorber, Josiah Hester, and Przemysław Pawełczak. Time-sensitive intermittent computing meets legacy software. In *Proceedings of the Twenty-Fifth International Conference on Architectural Support for Programming Languages and Operating Systems*, pages 85–99, 2020.

[11] Kiwan Maeng and Brandon Lucia. Adaptive low-overhead scheduling for periodic and reactive intermittent execution. In *Proceedings of the 41st ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 1005–1021, 2020.

[12] Eren Yildiz, Saad Ahmed, Bashima Islam, Josiah Hester, and Kasim Sinan Yildirim. Efficient and safe i/o operations for intermittent systems. In *Proceedings of the Eighteenth European Conference on Computer Systems*, pages 63–78, 2023.

[13] Eren Yildiz, Khakim Akhunov, Lorenzo Antonio Riva, Arda Goknil, Ivan Kurtev, and Kasim Sinan Yildirim. Adaptable runtime monitoring for intermittent systems. In *Proceedings of the Nineteenth European Conference on Computer Systems*, pages 1175–1191, 2024.

[14] Eren Yıldız, Lijun Chen, and Kasim Sinan Yıldırım. Immortal threads: Multithreaded event-driven intermittent computing on {Ultra-Low-Power} microcontrollers. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 339–355, 2022.

[15] Eren Yıldız and Kasım Sinan Yıldırım. Persistent timekeeping using harvested power measurements. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems*, pages 572–574, 2021.