

# A Control Flow for Transiently-Powered Energy Harvesting Sensor Systems

Domenico Balsamo, Oktay Cetinkaya, *Member, IEEE*, Alberto Rodriguez Arreola, Samuel C. B. Wong, Geoff V. Merrett, *Senior Member, IEEE* and Alex S. Weddell, *Member, IEEE*

**Abstract**—Transient computing enables application execution to be performed despite power outages. Although it handles the non-deterministic nature of energy harvesting (EH), sensor systems envisioned by the IoT seek more cost- and volume-effective solutions, which are better tailored to application requirements. Additionally, a major drawback of transient computing, keeping track of time, hinders its widespread adoption in the IoT. To overcome these challenges, this paper proposes a control flow for sensor systems by combining two state-of-the-art transient computing schemes in an energy-aware manner, underpinned by a strategy for timekeeping. It enables application execution to be reliably performed even under the most severe EH conditions, with an improved cost and volume efficiency, i.e., smaller energy storage. Benefiting from the combination of the two schemes, dynamic adjustment of system performance is achieved, while the time is accurately tracked. To illustrate the applicability of this flow to actual sensor systems, two case studies: a bicycle trip computer and a step counter, are presented. Empirical results reveal that, even with a tiny amount of energy harvested ( $\approx$  tens of  $\mu$ J), our proposed approach can meet application requirements with smaller storage, i.e., 40% and 66% reduction in required capacitance for the presented case studies.

**Index Terms**—Control Flow, Transient Computing, Energy Harvesting, Sensor Systems, Timekeeping, IoT Applications.

## I. INTRODUCTION

Emerging sensor systems in the IoT domain are mostly battery-powered, yielding a trade-off between lifetime and small form-factors. Thus, research has looked towards replacing batteries with energy harvesting (EH), which converts ambient sources into electricity [1]. Although EH provides infinite energy in theory, its output is highly-varying and intermittent, and thus may fail frequently. This can be rectified by a large energy storage, smoothing temporal dynamics of the source and load [2]. This is known as *energy-neutral* operation, which attempts to balance energy consumption against the harvested energy over a period of time [3]. However, this storage increases the system’s volume, weight, and cost, and takes longer to charge, which may delay the start-up.

A relatively new paradigm termed “transient computing” [4] enables computation to span power failures, whilst minimizing the need for large energy storage. Depending on the control strategy, transient computing offers *task-based* and *reactive*

schemes (Sec. II) [5]. In *task-based* schemes, the application is divided into small tasks (e.g., sensing, processing, and transmission), each of which has to be completed within one power cycle [6]. The results of a completed task and a snapshot of the system state for the transition to the next task are saved into a non-volatile memory (NVM). In *reactive* schemes, however, the tasks are not required to be executed in one cycle as the entire system state is saved upon power failure [7]. This enables task execution to be split across consecutive power cycles, minimizing the storage size. However, they are not suitable for tasks disallowing power failures, i.e., the tasks that cannot be split, such as sensing and transmission, due to causing unreliability. Even though task-based schemes ensure higher reliability in task execution, they require energy storage to be sized for the most energy-consuming task.

Task-based and reactive schemes have been considered separately in the literature, and there is as yet no work focusing on concurrent utilization of them. This can provide significant benefits, such as preserved reliability as well as increased cost and volume efficiency, by enabling tasks to be split when possible. These benefits can be granted by incorporating the EH process into the application model, enabling task-based and reactive schemes to be combined in an energy-aware manner. Thus, application execution can be performed even under the most severe EH conditions by dynamically adjusting the system performance, i.e., quality-of-service (QoS), according to EH output. Furthermore, as the intermittent supply may incapacitate traditional time-keeping methods, e.g., real-time clock (RTC), the fundamental challenge of transient systems, i.e., keeping track of time, has to be tackled.

We address the above-mentioned requirements through a control flow that merges the benefits of task-based and reactive schemes in an energy-aware manner and provides a strategy for timekeeping. The key contributions are as follows:

- a novel methodology combining task-based and reactive schemes to enable transient systems to operate with minimal energy storage without compromising reliability (Sec. III);
- an energy-aware approach linking the application execution to energy availability, i.e., dynamic adjustment of the QoS depending on the harvested energy;
- a strategy for timekeeping, i.e., using either one or the combination of applicable mechanisms: time-intertwined sensing, capacitor decay, and/or wireless transmission, depending on application and event attributes;
- two case studies: a wireless bicycle trip computer and a step counter, revealing the applicability of the proposed control flow to actual sensor systems (Sec. IV & Sec. V).

The authors are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK. e-mail: {db2a12, oc1y18, ara1g13, scbw1g19, gvm, asw}@ecs.soton.ac.uk.

D. Balsamo is also with the School of Electrical and Electronic Engineering, Newcastle University, Newcastle upon Tyne, UK. e-mail: ndb163@ncl.ac.uk.

This work was supported by the Mexican CONACYT and the UK Engineering and Physical Sciences Research Council (EPSRC) Platform Grant EP/P010164/1. Experimental data used in this paper can be found at DOI:10.5258/SOTON/D1364.

## II. RELATED WORKS

An EH sensor system adopting *task-based* operation is the wireless identification and sensing platform (WISP) [8], where devices energized by an RF source transmit their data back via backscattering. An RFID-based system called WISPCam [9], an EH wireless camera originated from WISP, is another example of this. Similarly, a UHF RFID chemical sensor array, proposed in [10], performs ambient sensing without a battery. These systems are triggered when the harvested energy is sufficient to complete the tasks with no interruptions.

Despite reducing the storage size compared to *energy-neutral* operation, task-based schemes may still require additional circuitry, such as MPPT and DC-to-DC converters [6], as the storage size is increased to complete the largest task. To cope with this issue, *reactive* schemes have been proposed, minimizing the storage size. Some are based on checkpoints inserted in the application program at compile-time, which check the input voltage to decide whether a snapshot has to be saved [7]. To avoid saving periodic snapshots, schemes that dynamically adapt the checkpoints in the application program, based on energy availability, have been developed [11], [12]. Alternatively, some [4], [13] monitor the input voltage and save a snapshot when it drops below a predefined threshold. However, with reactive schemes, the tasks that need to be completed without power interruptions, e.g., sensing and transmission, may fail making the sensor system unreliable.

In addition to above-mentioned challenges, an important functional requirement of the IoT applications is to have a sense of time for different purposes, e.g., to define whether a sampled parameter is still valid. Yet, tracking time with an RTC [14] is not an option for transiently-powered EH sensor systems due to the volatility of the source. Two methods have been proposed to address this issue [15]. *TARDIS* calculates the elapsed time based on SRAM decay; however, it is very inaccurate and power-hungry. *CusTARD* measures the time based on the discharge of a capacitor, whose size depends on the maximum time to be measured. This method is more accurate than *TARDIS* and requires simpler circuitry. Alternatively, sensor data by itself can provide a time reference, which is called time-intertwined sensing. For example, in [16] the duration of a bicycle trip is estimated by the division of sensing results, i.e., the traveled distance and the cycling speed.

In our previous studies [17], [16], we used *event detection sensors*, in which the EH source not only powers the system but also operates as a sensor, i.e., the system is powered by the parameter that is being sensed. [17] presents a step counter run by the energy harvested from human motion, where the energy presence indicates also that a step is taken. Similarly, [16] presents a bicycle trip computer, in which the energy harvested by the rotation of a wheel helps to estimate speed and distance while supplying the sensor system. However, these works did not adopt either a dedicated control flow adjusting the QoS based on the EH output or a timekeeping mechanism.

Therefore, this paper proposes a control flow for reliable application execution with the minimum storage possible by combining task-based and reactive schemes with a strategy for timekeeping.

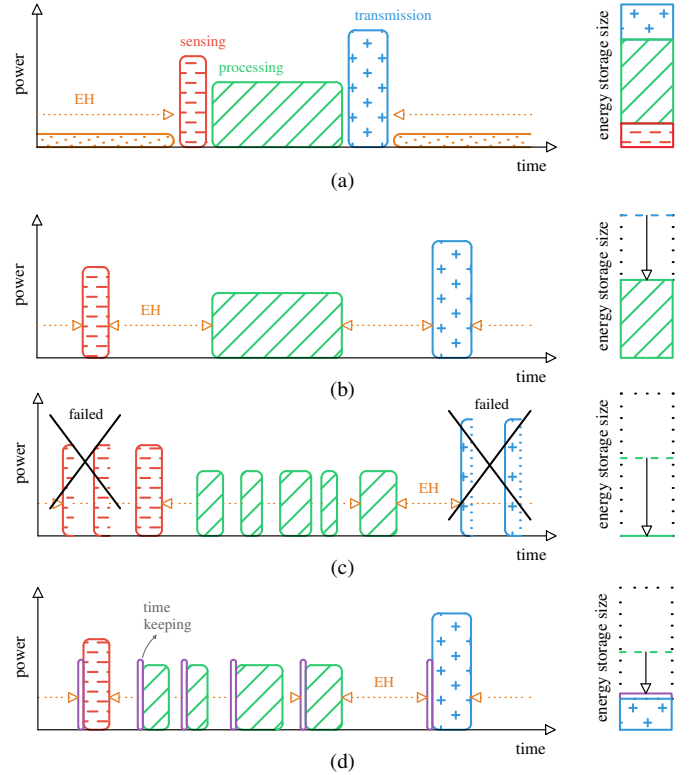


Fig. 1: Operational schemes: (a) energy-neutral; (b) task-based; (c) reactive; (d) our proposed control flow combining task-based and reactive schemes.

## III. PROPOSED CONTROL FLOW

Fig. 1 illustrates how tasks are performed depending on the EH process in different operational modes: (a) *energy-neutral*, (b) *task-based*, and (c) *reactive* schemes. As seen, moving from energy-neutral to task-based operation can considerably reduce the energy storage size. Reactive schemes, on the other hand, can make that storage even smaller (nearly zero) by enabling tasks to be split across consecutive power cycles. However, they cannot ensure forward progress in a reliable way, i.e., failed tasks as shown in Fig. 1(c). Considering these, we propose a control flow combining task-based and reactive schemes, Fig. 1(d), for reliable application execution, where the QoS is dynamically adjusted in an energy-aware manner with a minimum storage possible. It includes EH source profiling, adapts a strategy for timekeeping, and performs across three successive operational modes explained below.

### A. EH Source Profiling

As the first step of the proposed control flow, the EH source is profiled to determine the size of the energy storage needed as well as to establish the optimal triggering strategy.

Fig. 2(a) shows a sensor system directly coupled with a *small* capacitor  $C$ , which starts operating immediately after the voltage on capacitor,  $V_C$ , exceeds the minimum operating voltage,  $V_{min}$ . This configuration is suitable for systems operated by a *high-power* EH source, meaning that the harvester is able to supply enough power to directly sustain the system's operation. If the energy availability is due to the occurrence

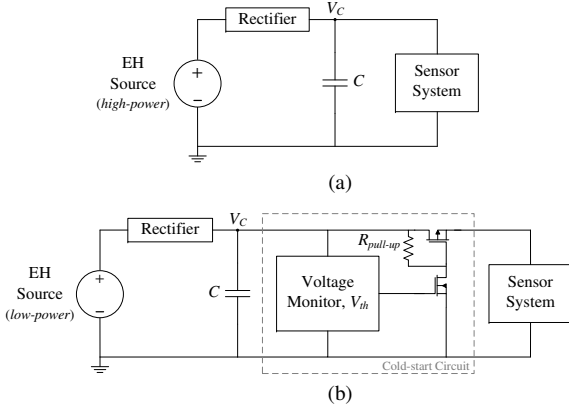


Fig. 2: Schematic of two EH sensor systems, powered by (a) a *high-power* source; (b) a *low-power* source.

of a particular event, it refers to an *event-based* triggering strategy. An example is a wind turbine, which usually supplies a large amount of energy but in short bursts.

The sensor system in Fig. 2(b), however, relies on a cold-start circuit (consisting of a voltage monitor, a pull-up resistor, and two switching MOSFETs), where the system is turned only on if the voltage on a *large* capacitor,  $V_C$ , is above a pre-defined threshold,  $V_{th}$ . This configuration is suitable for systems operated by a *low-power* EH source, meaning that the source is unable to supply enough power to directly sustain the system's operation. This refers to an *opportunistic* triggering strategy, where the computation is performed as soon as the buffered energy is sufficient. Small photovoltaic cells generating energy from diffused lights, i.e., indoor lighting, belong to this category.

In some cases, the system may harvest energy and information at the same time, i.e., *event detection sensors*, where the EH source becomes the de-facto sensor providing useful information related to the parameters being sensed. Thus, EH source profiling should consider not only the source's mere ability of powering the system but also its capability of meeting the application's sensing needs.

## B. Keeping Track of Time

Timekeeping in transient systems is a major challenge since the off periods between the events, i.e., the intervals with low or no energy availability, are usually non-deterministic and may vary from milliseconds to minutes. Depending on application specifications and the frequency of events, the methods depicted in Fig. 3 can be employed: (a) *time-intertwined sensing*, (b) *capacitor decay*, and (c) *wireless transmission*.

In some applications, events are directly correlated with time, i.e., the parameter being sensed also reveals the duration between the two consecutive events  $d_E$ , which ranges from milliseconds to second(s), Fig. 3(a). This method provides a good accuracy on the elapsed time from the first transmission (the 0 point) to the other,  $t_T$ , by tracking the duration between events without any additional component. However, if the events are not intertwined with time,  $d_E$  and so  $t_T$  can be estimated by capacitor decay, where  $d_E$  is typically in the order of seconds, Fig. 3(b). This method allows timekeeping with a slightly lower accuracy than the first method at the

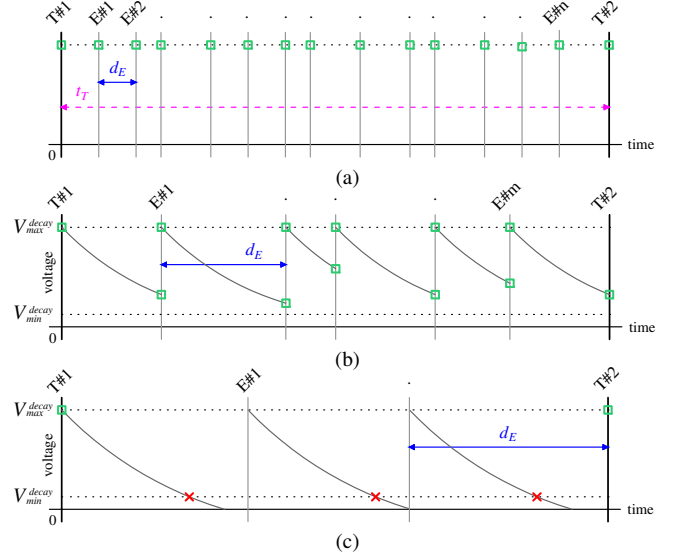


Fig. 3: Timekeeping mechanisms depending on event occurrence: (a) time-intertwined sensing; (b) capacitor decay; (c) wireless transmission. p.s. “E” is event and “T” is transmission.

cost of volume and redesign of the sensor system due to the addition of a capacitor,  $C_D$ . If  $d_E$  is ranging from multiple seconds to minute(s), however, capacitor decay will no longer be an option, since even the maximum size capacitor  $C_D^{max}$  allowed, in consideration of volume and charging time constraints, will not have enough voltage to sense the time when the next event occurs, Fig. 3(c). Such applications can use two consecutive wireless transmissions, where the received packets are timestamped. This will give  $t_T$  and hence a rough estimate of  $d_E$  with the lowest accuracy among all methods.

By taking these methods into account, our control flow by default adopts wireless transmission as the worst case scenario to provide some sense of time. However, depending on the event attributes, it allows applications to obtain a more accurate timekeeping mechanism by combining one of the other methods with the wireless transmission.

## C. Operation Architecture

Fig. 4 illustrates the operational architecture of our proposed control flow combining task-based and reactive schemes across three successive operational modes. Each mode has its own characteristics, and they perform by following the outcomes of the previous mode. To be more precise, Mode 1 depends on the outcomes of Mode 0, and Mode 2 is based on Mode 1.

When the system recovers from a power outage, i.e.,  $V_C$  exceeds  $V_{min}$  in the *high-power* source case ( $V_C > V_{min}$ ) or  $V_{th}$  in the *low-power* case ( $V_C > V_{th}$ ), basic operations, e.g., sensing for time keeping, are performed.

### In Mode 0,

- the system first checks the activity of interest to provide Mode 1 with some insights to adjust the QoS of the tasks;
- a *task-based* scheme is utilized to perform non-splittable tasks (e.g., sensing), where the outcomes of the tasks and the system state for the transition to Mode 1 are saved into the NVM;
- after Mode 0, the system moves to Mode 1.

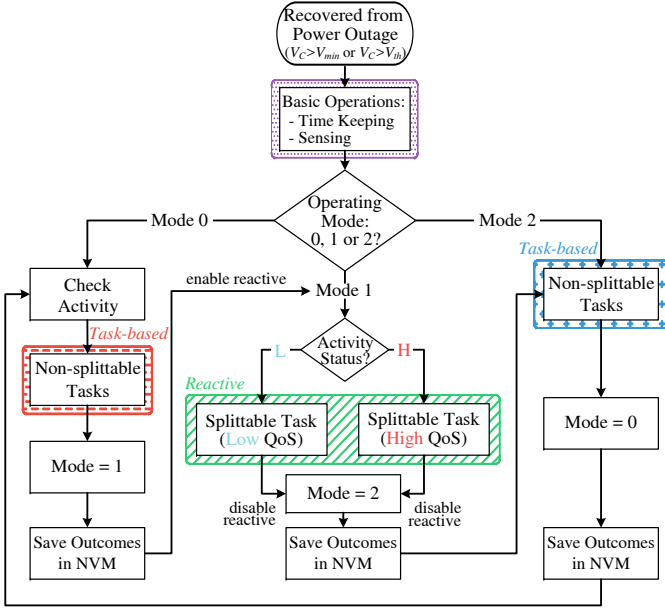


Fig. 4: Operational architecture of the proposed control flow.

#### In Mode 1,

- the system first checks the activity status provided by Mode 0 to decide between high or low QoS;
- it then uses a *reactive* scheme to perform splittable tasks (e.g., processing), where a snapshot is saved immediately before a power failure and restored when power returns;
- the outcomes of the tasks and the system state for the transition to Mode 2 are saved into the NVM;
- after Mode 1, the system moves to Mode 2.

#### In Mode 2,

- the system performs non-splittable tasks (e.g., transmission) in a *task-based* fashion;
- the outcomes of the tasks and the system state for the transition to Mode 0 are saved into the NVM;
- after Mode 2, the system moves back to Mode 0.

Here,  $C$  needs to buffer enough energy ( $V_C \geq V_{min}$  or  $V_C \geq V_{th}$ ) to at least complete the basic operations and execute Mode 0 or 2, depending on which requires more energy (in case of insufficient energy, the system waits for the next power cycle). In other words, the size of  $C$  is no longer determined by Mode 1, which was originally the most energy consuming operation, since the long-running computation is accomplished in the course of multiple cycles with less energy needs.

In consideration of specific challenges, this control flow is designed to be applicable to any transiently-powered EH sensor system. An example is the vibration level monitoring for predictive maintenance of marine engines, where the sensor acts also as a transducer, delivering data and power at the same time. Another example is a smart water meter, which extracts energy from temperature gradients to power the ultrasonic sensors measuring the flow rate. In this paper, we demonstrate our proposed flow through implementation and experimental evaluation of two case studies: a bicycle trip computer and a step counter, without loss of generality.

## IV. BICYCLE TRIP COMPUTER: A CASE STUDY

In this section, a transiently-powered bicycle trip computer employing the proposed control flow is presented. The system, supplied by a kinetic harvester, measures the cycling speed and the travelled distance as well as the elapsed time.

### A. EH Source Profiling

The harvester consists of a magnet attached to the wheel and a corresponding coil placed on the fork of the bicycle. The electromagnetic field induced by every rotation of the magnet generates an energy pulse at the coil, which is high enough to power the system without additional energy storage, i.e., it is a *high-power* source. Here, the energy pulse not only supplies electrical power but also acts as an *event detection sensor*, since its amplitude, and the frequency between consecutive pulses are directly proportional to the cycling speed.

The induced voltage  $V_{coil}$  is used to estimate the speed, travelled distance, and the elapsed time. Faraday's law describes this voltage as  $V_{coil} = -\partial\Phi_B/\partial t$ , where the magnetic flux  $\Phi_B$  depends on the speed of the wheel. Therefore, the peak voltage increases with increasing speed, while the duration of the pulse decreases. The pulse generated with each rotation powers the system, so is an *event-based* architecture.

### B. Adapted Operation Architecture

Fig. 5, indicates that the system first executes basic operations, i.e., a rough estimate of speed and time. Then, in Mode 0, it checks the activity via an ADC operation ( $ADC_1$ ) measuring the peak voltage of the pulse. If there is enough energy, however, a second ADC operation ( $ADC_2$ ) is performed by sampling the pulse profile, traces of both are shown in Fig. 6(a).

Afterward, the system moves to Mode 1, in which, by following the outcomes of Mode 0, cycling speed and time are estimated with various accuracies. The first option is to use the peak voltage only. Alternatively, both pulse profile and peak voltage can be utilized to provide more accurate

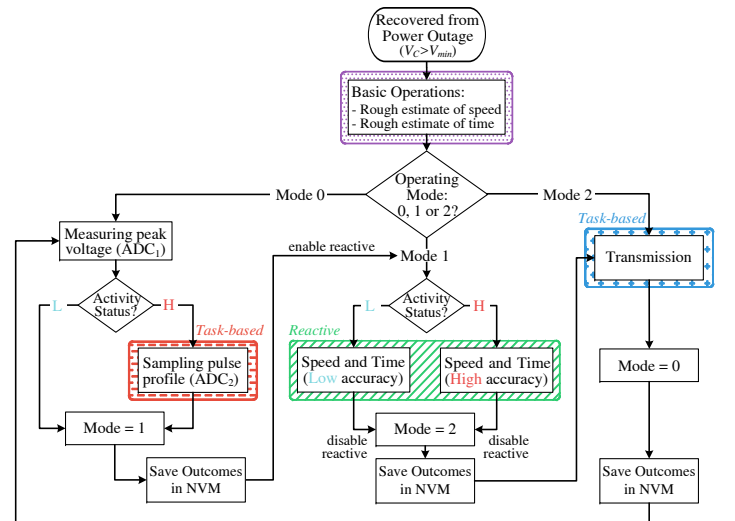


Fig. 5: Proposed control flow applied to the bicycle trip computer.

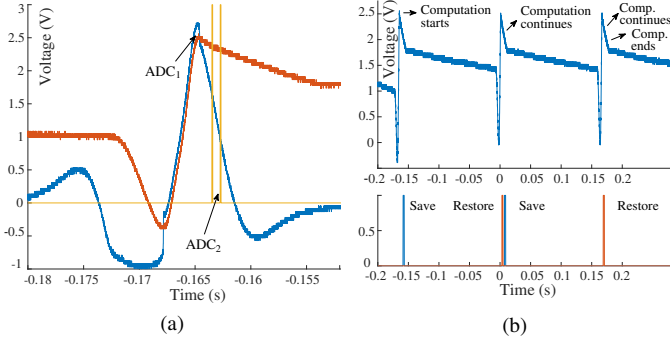


Fig. 6: Operation of the transiently-powered EH bicycle trip computer, (a) ADC<sub>1</sub> measures the peak voltage (orange trace), while ADC<sub>2</sub> measures the pulse profile (blue trace); (b) the long-running computation split into consecutive power cycles using [4].

results by using a Kalman filter. This filter is considered for the long-running computation in Mode 1 due to its ability in characterizing noise sources as well as minimizing their effect. The computation can then be split into consecutive power cycles as in Fig. 6(b), thanks to the reactive scheme adopted [4].

In Mode 2, outcomes of the long-running computation are wirelessly transmitted via a low-power protocol and saved into the NVM. Then, the system moves back to Mode 0.

### C. Keeping Track of Time

After EH source profiling and estimating the speed, the elapsed time is calculated based on the intrinsic relationship between speed and distance, which refers to *time-intertwined sensing*. The total distance is measured by the number of times that the system recovers from a power outage, i.e., every rotation of the wheel, multiplied by the circumference of the wheel. The total elapsed time between two consecutive speed estimations is then calculated by using this cycling speed.

### D. System Architecture and Energy Requirements

The trip computer is implemented in line with the architecture in Fig. 2(a). This comprises a low-power microcontroller unit (MCU), MSP430FR with a ferroelectric RAM (FRAM), and a 2.4GHz transceiver, NRF24L01+, for wireless communication. This module communicates with the MCU through SPI and performs data transmission via ShockBurst [18]. As expected, wireless communication in Mode 2 is the most energy consuming task (see Table I), which cannot be split across consecutive power cycles. Thus,  $C$  needs to buffer enough energy to complete the basic operations and execute

TABLE I: Current, time, and energy requirement of each operation.

Operation	Current (mA)	Time (ms)	Energy ( $\mu\text{J}$ )
ADC Measurement	0.704	0.07	0.82967
MCU Start-up	0.704	0.7	2.0165
Kalman Filter	0.704	7.24	14.087
Snapshot	0.704	1.865	1.988
Restore	0.704	1.805	2.9666
TX Configuration	1.3	2.1	6.5315
TX Transmission	14.5	0.15	4.5071



Fig. 7: Fabricated single-sided PCB of the bicycle trip computer.

Mode 2. This equates to a  $C$  of  $18.24\mu\text{F}$ , which is significantly smaller than the  $25.48\mu\text{F}$  required to complete the Kalman filter operation in Mode 1, i.e.,  $\approx 40\%$  reduction in size. As a result, a  $20\mu\text{F}$  capacitor, considering the commercially available components, is added to the platform. Since neither a large energy storage nor a power management circuit is required, the design is completely self-contained on a single sided  $2.0 \times 1.4\text{cm}^2$  PCB area, as shown in Fig. 7.

### E. Functional Validation and Comparative Evaluation

A speedometer is used to experimentally evaluate the operation of system. Fig. 8 shows the range of cycling speed and the corresponding harvested energy per each rotation, where the system moves from low to high accuracy when the harvested energy per rotation is higher than  $16\mu\text{J}$ . It means that the system can perform a second ADC by sampling the pulse profile, and estimate the cycling speed using the Kalman filter in Mode 1. The empirical measurements gathered by the speedometer reveal that the system achieves less than 1% error for speed, distance, and time in high accuracy mode, where the low accuracy mode has the average errors of 8%, 3%, and 5%, respectively.

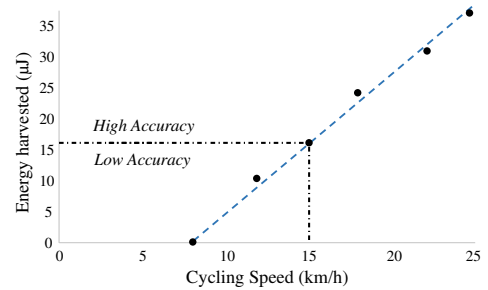


Fig. 8: Cycling speed vs. energy harvested.

## V. WIRELESS STEP COUNTER: A CASE STUDY

In this section, a transiently-powered wireless step counter employing the proposed control flow is presented. The system, supplied by a kinetic harvester, measures the step rate and the metabolic equivalent (MET) as well as the walking time.

### A. EH Source Profiling

The harvester is a 50-layer ferroelectret insole, which converts mechanical energy generated by the heel pacing on the floor into electricity [19]. This source is profiled by using three different capacitances,  $C$ , considering the decoupling

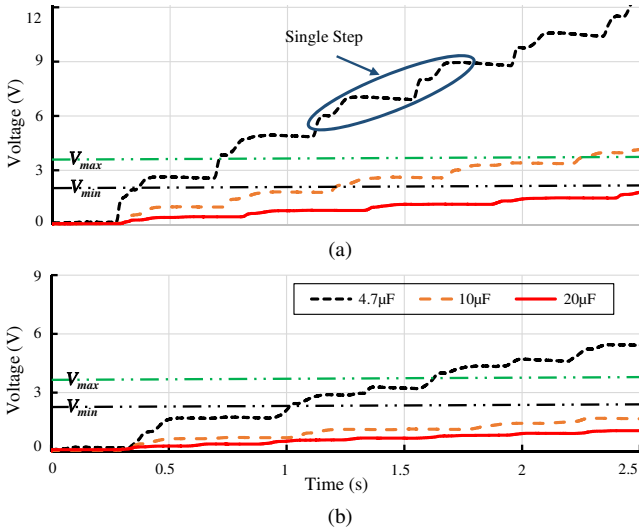


Fig. 9: Output voltage  $V_C$  for each capacitor, in (a) *Running* mode ( $\sim 120$  steps per minute); (b) *Walking* mode ( $\sim 70$  steps per minute).

capacitors commonly used in sensor systems [20]. Fig. 9 shows the voltage across these capacitors after each step in *running* ( $\sim 120$  steps per minute [21]) and *walking* mode ( $\sim 70$  steps per minute [21]). Here,  $V_{min}$  and  $V_{max}$  are the min. and max. operating voltages for a typical low-power MCU. The insole generates one positive (heel-strike) and one negative (toe-off) pulse at each step, which results in two increments in  $V_C$ . In running mode, the force applied to the insole is higher than in walking mode, causing a higher voltage increment at each step. This leads to uncertainty in the number of steps taken before the system becomes active, which is higher for larger  $C$ . As a result, the harvester is characterized as a *low-power* source, and since it needs to collect enough energy to exceed  $V_{th}$  to become active, the operation refers to an *opportunistic* triggering strategy. Here, the source acts also as an *event detection sensor*, where the energy availability indicates that steps have taken.

### B. Keeping Track of Time

From EH source profiling, it is not possible to estimate the elapsed time between steps, as the output voltage is affected by various factors, such as walking speed, user's height and weight. Thus, *CusTARD*, i.e., *decay of a capacitor*, is included in the system to keep track of time during the off periods.

Fig. 10 shows the circuit used for keeping track of time and its connection with the MCU. The capacitor  $C_{time}$  is charged by a GPIO pin through the resistor  $R_{char}$  and a Schottky diode to minimize reverse current. Thus,  $C_{time}$  is discharged through the resistor  $R_{disch}$  and the internal ADC measures the decrease in voltage during an off period. The size of  $C_{time}$  and the value of  $R_{disch}$  are calculated considering the longest off period when walking at the slowest speed [22].

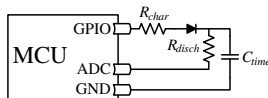


Fig. 10: Circuit used to track time during the off periods.

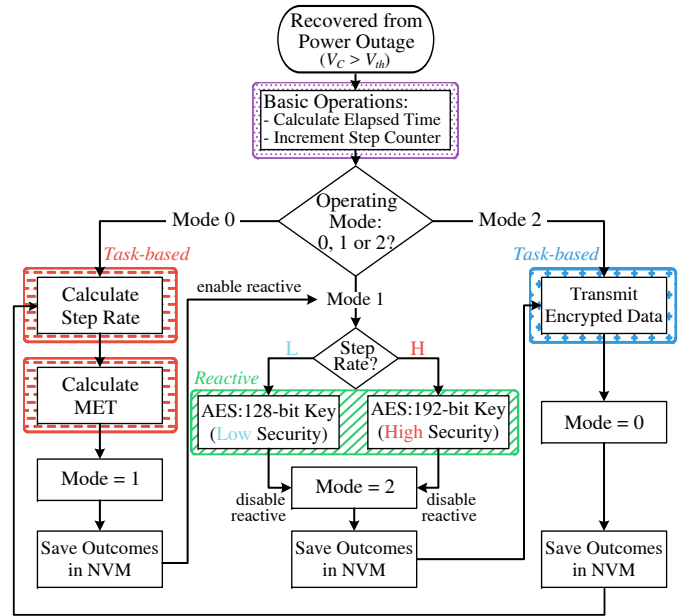


Fig. 11: Proposed control flow applied to the step counter.

This circuit requires a thorough characterization to define the non-linear behavior between the voltage decay on  $C_{time}$  and the elapsed time. This requires the measurement of the inverse current through the diode and the leakage current of  $C_{time}$  to compensate for their effect in the estimated time. The results are then saved into a look-up table (in the NVM).

### C. Adapted Operation Architecture

As shown in Fig. 11, the system first executes basic operations, i.e., estimate of the elapsed time between power cycles and the number of steps taken. Then, in Mode 0, it calculates the step rate, i.e., steps per minute (spm), and the MET, which is based on the step rate and the activity duration.

Afterwards, the system moves to Mode 1, in which, by following the outcomes of Mode 0, i.e., activity status, the data transmission security level is adjusted. The system encrypts the data packets with various security levels, low and high, based on the step rate estimated in Mode 0. If the step rate is less than 100spm, long-running computation, i.e., the advanced encryption standard (AES) algorithm, is executed with a 128-bit key. Otherwise, a 192-bit key requiring longer computational time is used. This computation is split across consecutive power cycles thanks to the reactive scheme adopted from [4].

In Mode 2, the encrypted data from Mode 1 are wirelessly transmitted via a low-power protocol and saved into the NVM. Then, the system moves back to Mode 0.

### D. System Architecture and Energy Requirements

The step counter is implemented considering the architecture in Fig. 2(b). This comprises the same MCU and transceiver as used in the first case study.

As expected, the most energy consuming task is again wireless communication (see Table II), which requires  $38\mu J$  of energy. Considering this, and the MCU voltage supply range,  $V_{min} = 2V$  and  $V_{max} = 3.6V$ , the minimum  $C$  needed is

TABLE II: Current, time, and energy requirement of each operation.

Operation	Current (mA)	Time (ms)	Energy ( $\mu$ J)
Encrypt	1.5	12.153	56.11
<i>CusTARD</i>	2.6	0.215	1.72
Count steps	0.9	0.092	0.26
MCU Start-up	1.5	2.232	10.32
TX Transmission	8.1	0.01	0.25
Snapshot/Restore	0.9	1.107	3.07
TX Configuration	3.66	2.436	27.41
Calculate Elapsed Time	1.15	0.277	0.98
Calculate Step Rate & MET	1.05	0.235	0.76

calculated as  $8.53\mu\text{F}$ . As a result, a  $10\mu\text{F}$  capacitor is added to provide a safety margin. On the contrary, if the longest task, i.e., encryption, is attempted to be completed without interruptions, the minimum  $C$  needed to complete this task would be  $14.16\mu\text{F}$ , which is 66% larger. Based on the flow in Fig. 11, the parameters in Table II, and the value of  $C$ ,  $V_{th}$  for this system is calculated as 3.4V. The demonstrator of the developed step counter is shown in Fig. 12.

### E. Functional Validation and Comparative Evaluation

A fitness band and a smartphone application is used to experimentally evaluate the operation of system. Fig. 14 shows the entire process from when the first step is taken to when the sensor system transmits the encrypted data. The system starts operating after four steps if there is not enough energy, i.e.,  $V_C < 1\text{V}$ , and needs only three steps for the following cycles.

To estimate the elapsed time, the voltage across  $C_{time}$  is measured just before  $V_C$  drops below  $V_{min}$  and immediately after it is restored. Mode 0 is then performed within a power cycle, while the data encryption is completed across three power cycles, in which the adopted reactive scheme saves and restores the system state consecutively.

The performance of our step counter is compared with a Mi band [23] and a fitness tracker smartphone application [24]. The experiments are performed by a person wearing three devices at the same time and performing aerobic activities in three different modes: walking, running, and walking-running. Each activity includes five repeats of 50 steps, i.e., 750 steps in total, where walking-running consists of 25 steps of walking and 25 steps of running. Fig. 13 compares our step counter and the other two solutions, where the experiments were performed ten times by a 70-kg person and the results were averaged.

The proposed solution has the average error of 13.2% in walking-running, 4.8% in running, and 8.8% in walking mode. These values for Mi Band are 9.6%, 13.6%, and 4.4%; and

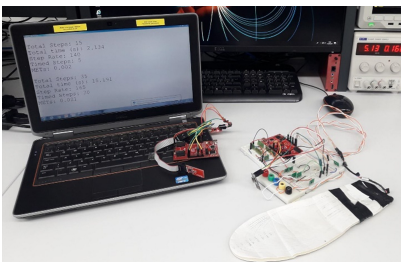


Fig. 12: Demonstrator of the developed step counter.

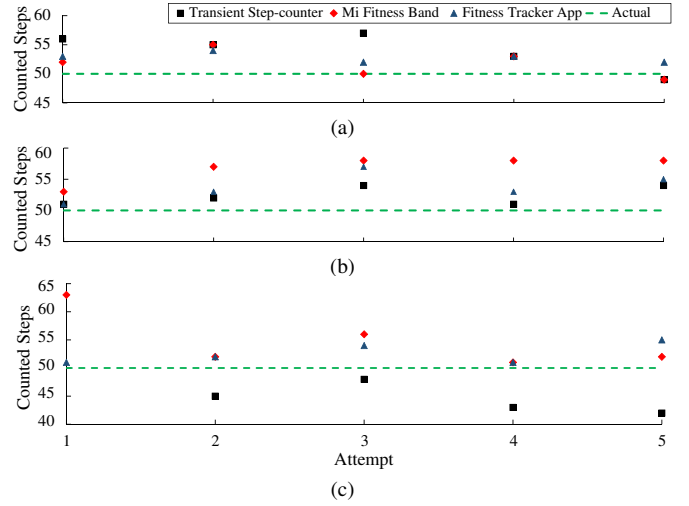


Fig. 13: Performance comparison of our step counter and the two other solutions for each activity: (a) walking; (b) running; (c) walking-running.

for the smartphone application are 5.2%, 7.6%, and 5.6%, respectively. The results show that our solution is comparable with off-the-shelf products, which are battery-powered and require much more energy than our proposal.

## VI. CONCLUSION

This paper presented a control flow for transiently-powered EH IoT systems, which combines *task-based* and *reactive* schemes in an energy-aware manner while providing a strategy for keeping track of time. It allows applications to be sustained with a smaller energy storage even powered by highly varying and scarce EH sources. The applicability of this proposal was validated through two case studies: a bicycle trip computer and a step counter. The experimental results showed that application requirements are met with an acceptable error while the energy storage size has reduced up to 66%.

## VII. ACKNOWLEDGEMENT

The authors would like to thank Dr. Sheng Yong, from the SEMS Research Group of the University of Southampton, for providing the 50-layer ferroelectret insole used in this work.

## REFERENCES

- [1] O. B. Akan *et al.*, "Internet of hybrid energy harvesting things," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 736–746, 2017.
- [2] A. Gomez *et al.*, "Precise, energy-efficient data acquisition architecture for monitoring radioactivity using self-sustainable wireless sensor nodes," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 459–469, 2017.
- [3] J. Hsu *et al.*, "Adaptive duty cycling for energy harvesting systems," in *Proc. of the Int. Sym. on Low power elect. and des.*, 2006, pp. 180–185.
- [4] D. Balsamo *et al.*, "Hibernus: Sustaining Computation During Intermittent Supply for Energy-Harvesting Systems," *IEEE Embed. Syst. Lett.*, vol. 7, no. 1, pp. 15–18, Mar. 2015.
- [5] S. T. Sliper *et al.*, "Enabling Intermittent Computing on High-performance Out-of-order Processors," in *Proc. 6th Int. Workshop Energy Harvesting & Energy-Neutral Sensing Syst.*, 2018, pp. 19–25.
- [6] A. Gomez *et al.*, "Dynamic energy burst scaling for transiently powered systems," in *Des., Auto. Test Europe Conf. Exh.*, 2016, pp. 349–354.
- [7] B. Ransford *et al.*, "Mementos: System Support for Long-running Computation on RFID-scale Devices," *SIGARCH Comput. Archit. News*, vol. 39, no. 1, pp. 159–170, Mar. 2011.

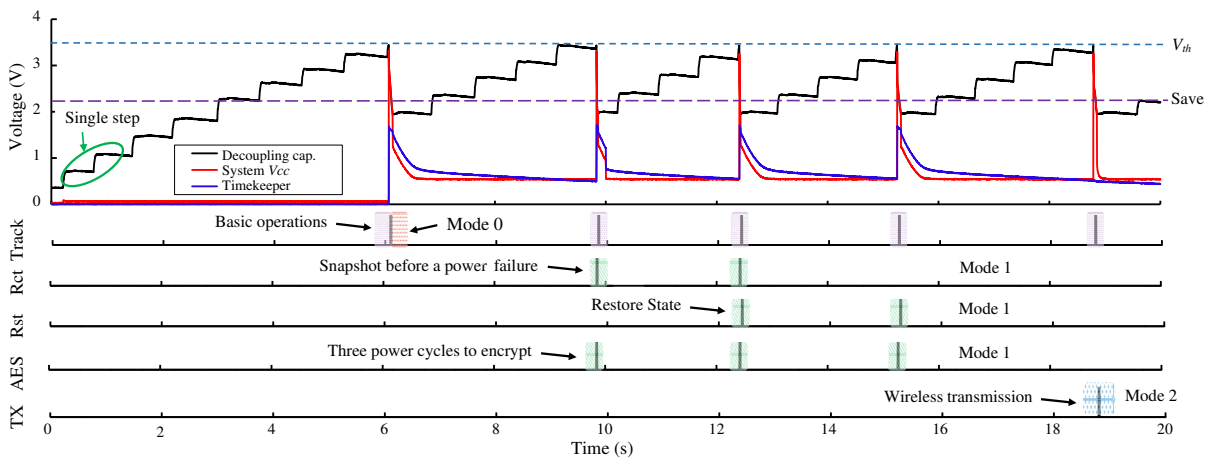


Fig. 14: Operation of the transiently-powered EH step counter.

- [8] A. P. Sample *et al.*, "Design of an RFID-Based Battery-Free Programmable Sensing Platform," *IEEE T. Instrum. and Meas.*, vol. 57, no. 11, pp. 2608–2615, Nov. 2008.
- [9] S. Naderiparizi *et al.*, "WISPCam: A battery-free RFID camera," in *2015 IEEE Int. Conf. RFID (RFID)*, Apr. 2015, pp. 166–173.
- [10] S. Manzari *et al.*, "Development of an UHF RFID Chemical Sensor Array for Battery-Less Ambient Sensing," *IEEE Sensors Journal*, vol. 14, no. 10, pp. 3616–3623, Oct. 2014.
- [11] J. V. D. Woude and M. Hicks, "Intermittent Computation without Hardware Support or Programmer Intervention," in *12th USENIX Symp. Operating Syst. Des. Implementation (OSDI 16)*, 2016.
- [12] K. Maeng and B. Lucia, "Adaptive Dynamic Checkpointing for Safe Efficient Intermittent Computing," in *13th USENIX Symp. Operating Syst. Des. Implementation (OSDI 18)*, 2018.
- [13] D. Balsamo *et al.*, "Hibernus++: A Self-Calibrating and Adaptive System for Transiently-Powered Embedded Devices," *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 35, no. 12, pp. 1968–1980, 2016.
- [14] J. Hester *et al.*, "Timely Execution on Intermittently Powered Batteryless Sensors," in *Proc. ACM Embed. Network Sensor Syst.*, 2017, pp. 1–13.
- [15] J. Hester *et al.*, "Persistent Clocks for Batteryless Sensing Devices," *ACM Trans. Embed. Comput. Syst.*, vol. 15, no. 4, pp. 1–28, Aug. 2016.
- [16] U. Senkans *et al.*, "Applications of Energy-Driven and Transient Computing: A Wireless Bicycle Trip Counter," in *ACM SenSys*, 2017.
- [17] A. A. Rodriguez *et al.*, "Intermittently-powered Energy Harvesting Step Counter for Fitness Tracking," in *IEEE Sens. App. Symp.*, 2017, pp. 1–6.
- [18] "Nordic Semiconductor nRF24L01+ Single Chip Datasheet. 2.4GHz ISM Transceiver," [Online Document], 2008. Available: [https://www.sparkfun.com/datasheets/RF/nRF2401rev1\\_1.pdf](https://www.sparkfun.com/datasheets/RF/nRF2401rev1_1.pdf).
- [19] S. Yong *et al.*, "Metal Layer reinforced multilayer ferroelectric-based energy harvester," *J. Phys.: Conf. Ser.*, vol. 1052, Jul. 2018.
- [20] "Texas Instruments MSP430FR5739 Datasheet," [Online], 2011. Available: <http://www.ti.com/lit/ds/symlink/msp430fr5739.pdf>.
- [21] C. Tudor-Locke *et al.*, "How fast is fast enough? walking cadence (steps/min) as a practical estimate of intensity in adults: a narrative review," *Br J Sports Med*, vol. 52, no. 12, pp. 776–788, 2018.
- [22] N. Sekiya *et al.*, "Optimal Walking in Terms of Variability in Step Length," *J. Orth. Sport. Phys. Ther.*, vol. 26, pp. 266–272, Nov. 1997.
- [23] "Mi Band 2," [Online], 2018. Available: <http://www.mi.com/en/miband2>.
- [24] "Fitness Tracker," [Online], 2018 Available: <http://rallysolutions.com>.