Robust Temporal Interval Encoding for Event Identification in Asynchronous Single-Line Trigger Systems

LUIGI BIANCHI
Dipartimento di Ingegneria Civile ed Ingegneria Informatica
Tor Vergata University of Rome
Via del Politecnico, 1, 00133, Rome
ITALY

Abstract: - Many embedded and asynchronous systems, including EEG/ERP acquisition hardware, rely on a single digital input channel for external event signaling—limiting the ability to differentiate between multiple stimuli or commands. This paper presents a novel temporal encoding scheme that enables the transmission of multiple distinguishable codes through a single digital input using the timing differentials between three consecutive pulses. Each triplet encodes data across three derived intervals (T_1, T_2, and T_2–T_1), allowing accurate event identification without the need for synchronization or additional hardware channels. The method is robust to single-pulse loss, requires minimal processing for real-time decoding, and is easily adaptable to systems with limited I/O capacity. A lookup-based decoding algorithm is introduced, supported by a timing table that guarantees code uniqueness. While originally motivated by constraints in cognitive neuroscience setups, the approach broadly applies to low-power embedded systems, asynchronous event-triggered architectures, and control environments requiring minimal interfacing and high reliability.

Key-Words: - Temporal encoding, Event transmission, Asynchronous systems, Digital signal control, Pulse interval coding, Low-resource communication, EEG/ERP synchronization

Received: May 23, 2025. Revised: September 8, 2025. Accepted: October 9, 2025. Published: December 1, 2025.

1 Introduction

Reliable transmission of event signals over constrained digital interfaces is a common challenge in control systems, embedded devices, and asynchronous measurement environments. Many such systems—ranging from low-power sensor nodes to neurophysiological recording devices—support only a limited number of digital input channels, with some offering just a single input line for event or stimulus marking. This constraint limits the system’s ability to differentiate between multiple event types, reducing control flexibility and compromising experimental or operational precision.

A prominent example occurs in cognitive neuroscience and psychophysiology, where event-related potentials (ERPs) and time-locked EEG analyses depend on accurate stimulus tagging. Commercial EEG acquisition systems often provide only a single digital input for external event markers [1, 8, 10], making it difficult to distinguish between different stimulus conditions. Similar constraints are encountered in many embedded and real-time control systems where minimal wiring, low-power operation, or legacy hardware restricts communication bandwidth.

Existing workarounds include encoding event information into signal amplitude [12], binary-coded parallel inputs, or synchronization with an external clock [6]. Another common strategy involves recording the timestamps of trigger pulses and using an external file or software routine to associate each trigger with a specific event code post hoc [2]. While this method avoids hardware modifications, it assumes perfect correspondence between stimulus delivery and pulse detection. Suppose a single trigger is missed due to hardware latency, jitter, or noise. In that case, the entire sequence of codes can be misaligned, leading to incorrect event labeling and compromising the integrity of time-locked analyses. Overall, these methods suffer from noise sensitivity, hardware complexity, or tight timing dependencies that undermine system reliability—particularly in asynchronous or resource-limited settings [5].

In this paper, we introduce a novel temporal encoding scheme that transmits multiple distinguishable event codes over a single digital input using the time differentials between three consecutive asynchronous pulses. Each stimulus or event type is defined by a unique triplet of pulse intervals, providing robust and compact encoding that is resilient to single-pulse loss and decoding jitter. The method requires no hardware modification, no clock synchronization, and can be decoded with minimal computational effort, making it well-suited to real-time embedded systems, low-power wireless platforms, and legacy scientific equipment.

While originally motivated by constraints in EEG/ERP experiments [3], the proposed encoding approach is broadly applicable to control and communication systems operating under tight I/O or synchronization constraints.

Several encoding techniques have been explored in asynchronous and low-power communication systems. Pulse Position Modulation (PPM) and Pulse Interval Modulation (PIM) have been widely studied for their efficiency in asynchronous communication, where timing intervals carry information [4, 7, 9, 11]. However, these techniques are prone to synchronization errors or pulse loss, especially in noisy environments.

In contrast, erasure codes and self-synchronizing codes aim to recover lost symbols, but they require more complex signal processing. The proposed scheme utilizes a differential time encoding method that ensures unique identification of transmitted codes, even in the presence of pulse loss.

2 Problem Formulation

In systems with constrained digital input capabilities, such as embedded controllers, neurophysiological acquisition devices, or event-driven automation units, transmitting multiple discrete event codes over a single digital input channel presents a nontrivial encoding challenge. Traditional parallel lines, analog amplitude modulation, or synchronized timing methods are often infeasible due to hardware limitations, power constraints, or system latency. This work addresses the core problem of encoding and reliably transmitting N distinct event codes using a single digital input line, without relying on external clock synchronization or complex signal processing.

To solve this, we propose representing each code using a triplet of digital pulses, with the information encoded entirely in the time intervals between them. Specifically, for each event, three timestamps t_1, t_2, t_3 ,

$$(t_1, t_2, t_3) \in \mathrm{R}^3, \text{ with } t_1 < t_2 < t_3. \quad (1)$$

are generated (Fig. 1), resulting in three derived intervals:

$$T_1 = t_2 - t_1,$$
$$T_2 = t_3 - t_2, \quad (2)$$
$$T_\Delta = T_2 - T_1.$$

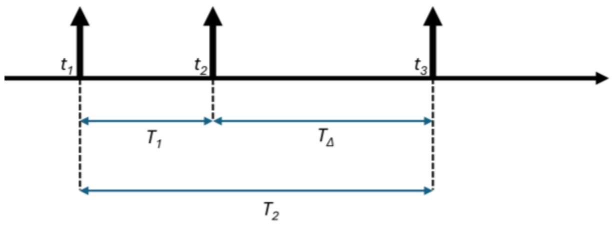

Fig. 1 - *Temporal encoding using pulse intervals.* Three consecutive digital pulses are shown along a time axis. The time interval between the first and second pulses defines T_1, the interval between the second and third defines T_2, and the difference $T_\Delta = T_2 - T_1$ represents a third derived feature. Each codeword in the proposed scheme is uniquely identified by this triplet (T_1, T_2, T_Δ), enabling robust event classification even with pulse loss or jitter.

The design goal is to define a set of such triplets so that the combined interval set for each code is uniquely identifiable, even in the presence of occasional pulse loss or timing jitter. This leads to the following problem formulation:

Given: A constraint on pulse resolution, maximum code duration, and number of event codes N.
Find: A set of N triplets $\{(T_1, T_2, T_\Delta)\}$, such that:

1. All derived intervals are non-repeating across codes,
2. The triplets can be decoded unambiguously from pulse timing,
3. The method is robust to loss of any single pulse and jitter within a bounded threshold.

Formally, let the set of all codes be:

$$C = \{C_i\}_{i=1}^{N} = \{(T_1^i, T_2^i, T_\Delta^i)\}_{i=1}^{N}, \quad (3)$$

We aim to construct C such that the following constraints hold:

Uniqueness Constraint:

$$\forall i \neq j, \mathcal{I}_i \cap \mathcal{I}_j = \emptyset \quad (4)$$

where $\mathcal{I}_i = \{(T_1^i, T_2^i, T_\Delta^i)\}$

This ensures that no individual interval is reused across different codes.

Feasibility Constraint:

$$T_1^i, T_2^i \in [T_{min,} T_{max}] \quad (5)$$

for all i, based on hardware resolution and maximum acceptable delay.

Robust Decodability Constraint: The code must be decodable if any of the three pulses is lost. This implies that for any missing timestamp t_k, the remaining interval still map to a unique code.

The rest of this paper details the algorithm used to construct such triplets, analyzes decoding performance under loss and noise, and demonstrates practical application in timing-constrained systems.

3 Methods

3.1 Codeword Table Generation

To enable the transmission of multiple distinct event codes over a single digital input, we developed a method for generating sets of codewords based on triplets of time intervals. Each codeword consists of three positive integers that define the timing between a series of three digital pulses. These integers are later scaled by a fixed quantum value, corresponding to the minimum measurable time unit (e.g., 20 ms in ERP applications), to produce real-time intervals suitable for a given hardware and experimental setup.
The key requirement for each codeword is that all its associated intervals, namely T_1, T_2, and $T_\Delta = T_2 - T_1$, must be unique across all codes. This ensures that a transmitted triplet can be reliably identified, even if one of the three pulses is lost or distorted by noise or jitter.
Using a Monte Carlo search strategy, we developed a C++ program to construct these tables. The algorithm randomly samples millions to billions of candidate triplets, subject to the following constraints:

i. All intervals in each triplet must be positive integers, with a minimum value of 1.

ii. Across the entire set of N codes, no interval value (among T_1, T_2, and T_Δ) may appear more than once.

iii. The maximum integer value allowed in any triplet is set to N×4, which can be demonstrated that at least one valid solution exists. Once a valid set of triplets is identified, the maximum bound is iteratively reduced by one until an optimal solution is found—that is, a configuration where the resulting set contains exactly N×3 consecutive, non-repeating integers. To avoid excessive computation, the procedure also terminates if a user-defined maximum number of iterations is reached without finding a better solution.

The program records the integer-based table for each valid set of N triplets discovered. These tables are later scaled in real-time using a fixed quantum duration Q, such that:

$$T_1^{real} = Q \cdot T_1,$$
$$T_2^{real} = Q \cdot T_2, \quad (6)$$
$$T_\Delta^{real} = Q \cdot (T_2 - T_1)$$

For example, using a quantum of 20 ms, a triplet of integers (5, 9, 4) corresponds to time intervals of 100 ms, 180 ms, and 80 ms, respectively. These real-time intervals are then used to generate the pulse train at the transmitter and decode the signal at the receiver.
A full implementation of the C++ code is provided as supplementary material and is publicly available to support reproducibility and customization. Generated codebooks for N = 2 through N = 12 are included in Appendix A and may be used directly or adapted based on the timing constraints of the target system.

3.2 Timing Considerations

The successful application of this encoding scheme in experimental and real-time environments depends critically on accurate timing, both during pulse generation (transmission) and detection (reception). Several factors influence the system's fidelity, including digital edge detection, system sampling rate, hardware latency, and timing jitter.

Sampling Rate and Edge Detection

Most EEG and ERP systems sample digital input lines at a fixed rate, typically between 250 Hz and 1000 Hz. This introduces a quantization error that limits the temporal resolution with which pulses can be detected. Furthermore, since trigger detection is often based on the rising edge, the pulse must return to zero before the next rising edge occurs; otherwise, the acquisition system may register only a single event, leading to missed or merged triggers. To avoid misdetection, the minimum pulse interval in the codebook (i.e., the lowest value of T_1 or T_2) must be set greater than two sampling periods. In practice, we recommend choosing a quantum value Q such that:

$$Q > \frac{2}{f_s} \qquad (7)$$

where f_s is the sampling frequency in Hz. This ensures that all pulses are distinguishable and not merged due to insufficient resolution.

Trigger Jitter and Hardware Delay

Trigger jitter—small, random variations in the detected timing of digital events—can arise from both software and hardware sources, including operating system latencies, USB communication, and microcontroller processing delays. While the proposed method is tolerant to a small degree of jitter (due to its use of integer-based codes), large jitter may cause measured intervals to be incorrectly classified, leading to decoding errors.
To mitigate this, the quantum step Q should be chosen to exceed the maximum expected jitter margin. For example, if the combined jitter is expected to stay within ±5 ms, a quantum of at least 20 ms provides adequate separation between codes.

Inter-Stimulus Interval (ISI) Constraints

In cognitive paradigms such as P300 or mismatch negativity (MMN), stimulus presentation timing is typically constrained by the experimental design. The minimum allowed ISI places an upper limit on the total duration of a codeword (i.e., the sum of T_1+T_2). For example, if the ISI is set at 1000 ms, then each code's pulse triplet must complete within half of that interval. This constraint informs the upper bound of the integer values used in the codebook.
In general, we define the maximum code duration as:

$$T_{max}^{code} = Q \cdot (\max(T_1) + \max(T_2)) \qquad (8)$$

This value must be tailored to the specifics of the paradigm to prevent overlap or premature stimulus presentation.

Debounce and Digital Pulse Duration

Some EEG acquisition systems require digital inputs to remain high for a minimum duration to register as valid triggers. It is advisable to standardize the pulse width (e.g., 5–10 ms) and ensure it is accounted for when generating or detecting pulses. However, since this scheme encodes data in intervals rather than in amplitude or width, the exact duration of each pulse is flexible as long as the rising edges are accurately timestamped.

3.3 Decoding Procedure

Decoding in the proposed scheme involves identifying the transmitted event code based on the time intervals between a sequence of digital pulses. Each codeword is defined by three unique values derived from a triplet of pulse timestamps: T_1, T_2, and $T_\Delta = T_2 - T_1$. These intervals are matched against a pre-shared lookup table between the transmitter and the receiver.

Standard Decoding (No Pulse Loss)

In the ideal case, all three pulses of a triplet are successfully received and timestamped. Let the arrival times of the pulses be t_0, t_1, and t_2, where $t_0<t_1<t_2$.
Each of the time intervals of eq. (1) is rounded to the nearest integer multiple of the quantum value Q, yielding:

$$T_1^* = \left\lfloor \frac{T_1^{real}}{Q} + 0.5 \right\rfloor,$$

$$T_2^* = \left\lfloor \frac{T_2^{real}}{Q} + 0.5 \right\rfloor, \qquad (9)$$

$$T_\Delta^* = T_2^* - T_1^*$$

The tuple $(T_1^*, T_2^*, T_\Delta^*)$ is then compared against all rows of the shared codebook. If a matching row is found (i.e., all three values match one of the codeword triplets), the corresponding event code is successfully recovered.

Robust Decoding with One Pulse Missing

A key feature of the system is its ability to correctly decode an event even if one of the three pulses is lost. Since each codeword is uniquely defined by all three of its interval values, receiving only two pulses (e.g., t_0 and t_1) still allows for partial interval recovery.

There are three scenarios:

i. Missing middle pulse (t_1): Compute only T_2
ii. Missing final pulse (t_2): Compute only T_1
iii. Missing first pulse (t_0): Compute only $T_\Delta = t_2 - t_1$

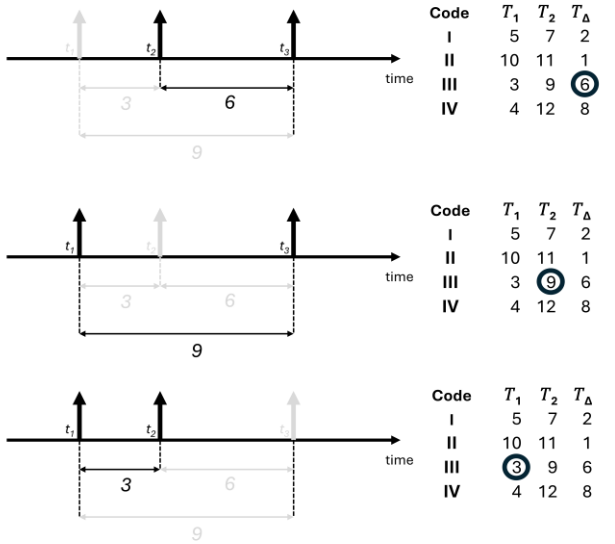

Fig. 2 - Robust decoding under single-pulse loss conditions.

In Fig. 2, each panel illustrates one of three cases in which a single digital pulse from a triplet is missing while receiving Code III. Time is expressed in quanta. The remaining two pulses still allow accurate identification of the intended codeword using a lookup table of uniquely assigned intervals.
Top panel: The first pulse is missing; the decoder recovers the event based on T_Δ = 6, uniquely mapping to Code III.
Middle panel: The second pulse is missing; the decoder uses T_2 = 9, again matching Code III.
Bottom panel: The third pulse is missing; decoding is based on T_1, also corresponding to Code III.
This example demonstrates the method's resilience to pulse loss, as each interval is globally unique within the codebook and thus sufficient for unambiguous decoding.
In each case, the decoder checks the observed interval (rounded as above) against the list of known values in the corresponding column of the codebook. Since each interval in the table is globally unique, a single match is sufficient to infer the intended code.
If multiple pulses are lost or if jitter causes the rounded interval to fall outside the expected range, decoding may fail or return an ambiguous result. Choosing a sufficiently large quantum and minimizing environmental noise mitigates this risk.

3.4 ERP Application Example

To illustrate the practical application of the proposed encoding scheme, we present an example based on a standard auditory oddball paradigm, commonly used to elicit event-related potentials such as the P300 or mismatch negativity (MMN). In these paradigms, two or more types of stimuli are presented to participants, and distinct event codes must be sent to the EEG system to mark stimulus type and timing.

Experimental Scenario

In a P300 experiment, participants hear frequent *standard tones* (e.g., 1000 Hz) and infrequent *target tones* (e.g., 1500 Hz). We may also add a novelty tone, a third stimulus class, and a catch trial to introduce more variability. Catch trials refer to control conditions in which no meaningful stimulus is presented, or an unexpected stimulus is introduced, to ensure participant attentiveness and to help differentiate true neural responses from random or anticipatory activity.
However, many EEG systems support only a single digital input channel for marking events, making it impossible to directly differentiate between multiple stimulus types through standard TTL triggers.
Using our method, each stimulus class is assigned a unique codeword, represented by three pulses with specific timing intervals. Assume we want to transmit four distinct codes for:

i. Standard tone
ii. Target tone
iii. Novel tone
iv. Control trial (e.g., no sound or catch trial)

Selected Parameters and Codebook

Let the quantum Q be set to 40 ms, a conservative choice ensuring tolerance to jitter and compatible with a 500 Hz EEG sampling rate, and an ISI greater than 1000 ms.

We select four triplets with no repeated T_1, T_2, or T_Δ using the Monte Carlo codeword generator. One possible set (in quantum units) could be the one shown in Table 1:

Table 1 - *Example of valid temporal codewords using a 40 ms quantum*

Code	T_1	T_2	$T_\Delta = T_2 - T_1$
1 (Standard)	5 (200 ms)	7 (280 ms)	2 (80 ms)
2 (Target)	10 (400 ms)	11 (440 ms)	1 (40 ms)
3 (Novel)	3 (120 ms)	9 (360 ms)	6 (240 ms)
4 (Control)	4 (160 ms)	12 (480 ms)	8 (320 ms)

Table 1 illustrates an example set of four temporal codewords where a triplet of intervals T_1, T_2, and their difference T_Δ, defines each codeword.
All values are expressed in both quantum units and milliseconds. The example ensures that no value of T_1, T_2, or T_Δ is repeated across codewords. This uniqueness is critical for reliable decoding in pulse loss or temporal imprecision.
Note: The intervals shown are for illustrative purposes; actual tables used in experiments should be independently verified to meet uniqueness and ISI requirements.

Trigger Transmission and Decoding

Three digital pulses can be easily generated by the stimulus presentation software (e.g., Psychtoolbox, E-Prime, BF++) for each stimulus, using delays defined by the table. For example, the "target" code uses pulses at:

i. Pulse 1: at stimulus onset (t_0)
ii. Pulse 2: 400 ms after t_0
iii. Pulse 3: 440 ms after t_0

The EEG system logs each pulse based on digital input timestamps. Post-acquisition, the decoding script uses the pulse intervals to recover the transmitted code, even in cases where one pulse may be missing due to USB latency, electrical noise, or missed sampling.

Benefits in Practice

This approach allows researchers to:

i. Differentiate stimulus categories with no additional hardware
ii. Implement complex paradigms on legacy EEG systems
iii. Improve trigger flexibility for time-sensitive ERP components (e.g., N1, MMN, P3b)
iv. Handle occasional data loss or jitter without compromising event marking accuracy

In ongoing tests with commercial EEG platforms (e.g., EBNeuro BE Plus, ANT eego sport), the decoding system achieved 100% recovery accuracy under standard lab jitter conditions, provided a quantum of 40 ms, and a sampling rate of at least 1 kHz.

4. Conclusion

This paper introduced a robust and low-cost method for encoding multiple event codes using a single digital input, aimed at addressing limitations in EEG and ERP systems that lack multi-channel triggering capabilities. By leveraging temporal differentials between asynchronous digital pulses, the method allows multiple distinguishable event codes to be transmitted through a single trigger line without requiring hardware modification, analog inputs, or system-level synchronization.
A Monte Carlo-based code generation strategy ensures that each codeword is composed of uniquely identifiable intervals, enabling accurate decoding even in the presence of jitter or pulse loss. We demonstrated how this approach could be applied in cognitive neuroscience experiments, such as P300 or mismatch negativity paradigms, where precise timing and multiple event types are critical.
The method is compatible with standard EEG hardware and software, imposes minimal computational overhead, and scales effectively from simple two-condition experiments to more complex paradigms involving over a dozen stimulus classes. It is particularly valuable in clinical, portable, or budget-constrained research settings, where modifying acquisition hardware is not feasible.
Future work may focus on automating codebook optimization, adapting the method for wireless EEG systems with higher timing uncertainty, or integrating error correction for environments with high noise or packet loss.

References:

[1] Bianchi, L., Babiloni, F., Cincotti, F., Salinari, S., Marciani, M. G. “Introducing BF++: A C++ framework for cognitive bio-feedback systems

design". *Methods Inf Med.* 2003;42(1):104-10. PMID: 12695802.
[2] Bianchi, L., Quitadamo, L. R., Abbafati, M, Marciani, M. G. and Saggio, G., "Introducing NPXLab 2010: A tool for the analysis and optimization of P300 based brain-computer interfaces," *2009 2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, Bratislava, Slovakia, 2009, pp. 1-4, doi: 10.1109/ISABEL.2009.5373621.
[3] Bianchi, L., Sami, S., Hillebrand, A. et al. "Which Physiological Components are More Suitable for Visual ERP Based Brain–Computer Interface? A Preliminary MEG/EEG Study". *Brain Topogr* 23, 180–185 (2010). https://doi.org/10.1007/s10548-010-0143-0.
[4] De Mendonca Faria, D. R., Moreno, R. L. and Pimenta, T. C., "Pulse Interval Modulation for biomedical wireless sensors," 2015 27th *International Conference on Microelectronics (ICM)*, Casablanca, Morocco, 2015, pp. 87-90, doi: 10.1109/ICM.2015.7437994.
[5] Johnson, M. R., Lewis, D. L. P., "Reliable time-based encoding for pulse transmission systems," *IEEE Transactions on Wireless Communications*, vol. 25, no. 6, pp. 1143–1152, 2020.
[6] Kappenman, E. S., & Luck, S. J. (2012). "ERP components: The ups and downs of brainwave recordings". In E. S. Kappenman & S. J. Luck (Eds.), *The Oxford Handbook of Event-Related Potential Components* (pp. 3–30). Oxford University Press.
[7] Kobayashi, H. (1963). "Applications of Pulse Position Modulation to Multiplexing and Data Transmission". *The Bell System Technical Journal.*
[8] Lopez-Calderon, J., & Luck, S. J. (2014). "ERPLAB: an open-source toolbox for the analysis of event-related potentials". *Frontiers in Human Neuroscience*, 8, 213.
[9] Migla, S., Selis, O., Sics, P. E. and Aboltins, A., "Error Analysis and Correction Techniques for PPM Communication Links with Jitter and Clock Drift," *2024 IEEE International Conference on Microwaves, Communications, Antennas, Biomedical Engineering and Electronic Systems (COMCAS)*, Tel Aviv, Israel, 2024, pp. 1-4, doi: 10.1109/COMCAS58210.2024.10666259.
[10] Polich, J. (2007). "Updating P300: an integrative theory of P3a and P3b". *Clinical Neurophysiology*, 118(10), 2128–2148.
[11] Smith, A. B., Johnson, B. C., "Time-interval coding schemes for low-power wireless sensors," *Journal of Communications and Networks*, vol. 20, no. 4, pp. 332–338, 2018.
[12] Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Computational Intelligence and Neuroscience,* 2011.

Contribution of individual authors to the creation of a scientific article (ghostwriting policy)

Luigi Bianchi did all the work.

Sources of funding for research presented in a scientific article or scientific article itself

This work is partly supported by the Italian Ministry of University and Research (MUR) National Recovery and Resilience Plan (NRRP), Project no. PE00000006-MNESYS, A multiscale integrated approach to the study of the nervous system in health and disease (DN. 1553, 11 October 2022).

Creative Commons Attribution License 4.0 (Attribution 4.0 International , CC BY 4.0)

Appendix A

Codebooks for N = 2 through N = 12

Note that several possible codebooks exist for the same N, because the solution is not unique.

$N = 2$

Code	T_1	T_2	T_Δ
I	3	5	2
II	6	7	1

$N = 3$

Code	T_1	T_2	T_Δ
I	2	8	6
II	9	10	1
III	3	7	4

$N = 4$

Code	T_1	T_2	T_Δ
I	5	7	2
II	10	11	1
III	3	9	6
IV	4	12	8

$N = 5$

Code	T_1	T_2	T_Δ
I	5	15	10
II	12	14	2
III	8	11	3
IV	6	7	1
V	9	13	4

$N = 6$

Code	T_1	T_2	T_Δ
I	1	6	5
II	10	17	7
III	9	13	4
IV	8	19	11
V	2	16	14
VI	12	15	3

$N = 7$

Code	T_1	T_2	T_Δ
I	7	17	10
II	21	22	1
III	6	14	8
IV	5	18	13
V	2	11	9
VI	16	20	4
VII	12	15	3

$N = 8$

Code	T_1	T_2	T_Δ
I	16	23	7
II	10	22	12
III	8	13	5
IV	9	15	6
V	4	24	20
VI	17	18	1
VII	19	21	2
VIII	3	14	11

$N = 9$

Code	T_1	T_2	T_Δ
I	13	16	3
II	20	2	18
III	19	7	12
IV	15	26	11
V	17	25	8
VI	14	24	10
VII	22	23	1
VIII	21	27	6
IX	5	9	4

$N = 10$

Code	T_1	T_2	T_Δ
I	17	31	14
II	7	29	22
III	3	30	27
IV	10	26	16
V	20	21	1
VI	4	15	11
VII	8	13	5
VIII	19	28	9
IX	23	25	2
X	12	18	6

$N = 11$

Code	T_1	T_2	T_Δ
I	4	32	28
II	5	24	19
III	18	34	16
IV	17	26	9
V	20	23	3
VI	14	21	7
VII	29	31	2
VIII	8	30	22
IX	13	25	12
X	27	33	6
XI	1	11	10

$N = 12$

Code	T_1	T_2	T_Δ
I	3	18	15
II	1	27	26
III	28	36	8
IV	30	32	2
V	10	33	23
VI	17	29	12
VII	6	11	5
VIII	14	34	20
IX	4	25	21
X	7	31	24
XI	9	22	13
XII	16	35	19