The Greek-Hanke-Henry Permanent Calendar

Dennis WK Khong*[1]

[1]Multimedia University, Malaysia

January 3, 2026

Abstract

The Gregorian calendar, while universally adopted, suffers from significant irregularities that impede economic calculation, software development, and administrative planning. The Hanke-Henry Permanent Calendar (HHPC) offers a mathematically superior alternative by stabilizing the 364-day year into fixed 30-30-31 day quarters. However, the adoption of the HHPC faces resistance due to the confusion resulting from using Gregorian month names for a non-Gregorian system. This paper proposes the Greek-Hanke-Henry (GHH) Permanent Calendar, a modification that replaces Latinate month names with the Greek alphabet (Alpha through Mu). This nomenclature shift ensures global neutrality, eliminates ambiguity in date parsing, and facilitates software internationalization (i18n). A Python implementation of the GHH system demonstrates its compatibility with the ISO-8601 standard and its superior handling of the "Omega Week" leap system. The source code is available at `https://github.com/denniskhong/ghh-calendar`.

1 Introduction

The measurement of time is fundamental to civilization, yet the current global standard, the Gregorian calendar, is an administrative relic of the Roman Empire. Its irregular month lengths (ranging from 28 to 31 days) and the drift of weekdays relative to dates create annual inefficiencies estimated in the billions of dollars due to rescheduling costs and software complexity (Hanke and Henry, n.d.).

While the **Hanke-Henry Permanent Calendar (HHPC)** (Hanke and Henry, n.d.) successfully resolves the structural irregularities of the year, it retains the traditional names (January, February, etc.). This creates a significant user interface (UI) problem: a date such as "January 31" exists in the Gregorian calendar but might not exist or falls on a different day in the HHPC. This ambiguity hinders adoption.

The **Greek-Hanke-Henry (GHH) Permanent Calendar**, as proposed, couples the structural stability of the HHPC with a distinct and neutral nomenclature based on the Greek alphabet.

Given the entrenchment of the Gregorian system, it is suggested that a "hard reform"—a sudden, universal switch—is neither necessary nor practical. Instead, the transition can be achieved through a **parallel calendar system**. In the digital age, calendars are fundamentally information layers within software; modern devices can effortlessly compute and display multiple date systems simultaneously. This dual-usage model is already the standard in many parts of the world: Muslim-majority nations frequently utilize the Hijri calendar alongside the Gregorian for religious and administrative purposes, while communities in East Asia integrate the Chinese or Buddhist lunar calendars into daily life. By implementing the GHH as an optional, parallel

*Corresponding author: `wkkhong@mmu.edu.my`. ORCiD: `https://orcid.org/0000-0002-9730-0944`.

standard within software operating systems, economic actors can voluntarily adopt its efficiencies for financial and administrative calculations without requiring an immediate upheaval of civil or religious traditions.

1.1 Barriers to Calendar Reform

Despite the mathematical and administrative advantages of a fixed calendar, purely scientific proposals face significant inertia from religious and societal institutions. From a religious perspective, the primary objection to calendar reform has historically been the disruption of the continuous seven-day weekly cycle, which is essential for the Sabbath observance in Abrahamic faiths. Previous attempts, such as the *World Calendar* proposed to the UN in the 1930, failed largely because they utilized "unaccounted days" (days outside the weekly cycle) to balance the year, which was unacceptable to religious communities (Bushell, 1961; Davies et al., 1999).

From an economic perspective, resistance stems from the high "switching costs" of legacy infrastructure. The modern financial system relies heavily on day-count conventions to calculate interest on bonds and mortgages. While a permanent calendar would eliminate the need for these complex conventions and reduce errors, the immediate cost of reconfiguring global IT systems and rewriting legal contracts creates a "switching cost" that locks the Gregorian calendar in place, similar to the QWERTY keyboard standard (David, 1985).

Therefore, to be viable, any calendar reform must retain the continuous seven-day week and the twelve-month year.

2 The Hanke-Henry Permanent Calendar

The core logic of the HHPC is that the year is divided into four identical quarters, each consisting of exactly 91 days (13 weeks).

$$Q_n = 30 + 30 + 31 = 91 \text{ days}$$

$$\text{Year} = 4 \times Q_n = 364 \text{ days}$$

Since 364 is divisible by 7, every specific date falls on the same day of the week, every year, forever.

Since the fixed calendar contains only 364 days, an extra week is added after the 12th month every five or six years to compensate for the accumulating drift relative to the astronomical year.

3 The Greek-Hanke-Henry Permanent Calendar

3.1 The Greek Nomenclature

To distinguish the GHH from the Gregorian calendar, the Roman-derived names are replaced with the Greek alphabet. This serves three purposes:

1. **Disambiguation:** "Alpha 15" cannot be confused with "January 15".
2. **Global Neutrality:** Greek letters are standard in mathematics and science.
3. **Sorting Efficiency:** The sequence Alpha–Mu implies a logical order.

Table 1: The Greek-Hanke-Henry Month Structure

Month Order	GHH Name	Days	Fixed Start Day
1	Alpha	30	Monday
2	Beta	30	Wednesday
3	Gamma	31	Friday
4	Delta	30	Monday
5	Epsilon	30	Wednesday
6	Zeta	31	Friday
7	Eta	30	Monday
8	Theta	30	Wednesday
9	Iota	31	Friday
10	Kappa	30	Monday
11	Lambda	30	Wednesday
12	Mu	31	Friday

3.2 Leap Logic: The Omega Week

The solar year is approximately 365.2422 days. A fixed 364-day calendar drifts by about 1.25 days per year. The Gregorian calendar corrects this by adding a leap day (February 29) every four years, which disrupts the weekly cycle.

The GHH, following the Hanke-Henry proposal, preserves the integrity of the 7-day week by accumulating the drift. Instead of a leap day, a **Leap Week** (7 days) is added to the end of the year roughly every 5 to 6 years. This week is named **Omega**. An Omega Week is inserted 6 years after the previous one, unless the 5th year ends on a Thursday (or Friday in a leap year), in which case it is inserted then.

The GHH synchronizes with the **ISO-8601** standard (ISO, 2019). "Alpha 1" is defined as the Monday of ISO Week 1. When the ISO year contains 53 weeks, the GHH calendar appends the Omega week.

4 Python Implementation

A Python algorithm to generate the GHH calendar and convert Gregorian dates is developed with the help of Gemini 3 Pro. The system utilizes `datetime.isocalendar()` to ensure robust handling of the leap week logic.

```python
def gregorian_to_ghh(self, date_obj):
    """Converts Gregorian date to GHH Date"""
    iso_year, iso_week, iso_day = date_obj.isocalendar()

    if iso_week == 53:
        return {
            "year": iso_year,
            "month": self.month_names[12], # Omega
            "day": iso_day,
            "weekday": self.week_days[iso_day - 1],
            "is_omega": True
        }

    day_of_year_idx = (iso_week - 1) * 7 + (iso_day - 1)
    days_sum = 0

    for i in range(12):
        length = self.month_structure[i]
```

```python
        if day_of_year_idx < days_sum + length:
            day_in_month = day_of_year_idx - days_sum + 1
            return {
                "year": iso_year,
                "month": self.month_names[i],
                "day": day_in_month,
                "weekday": self.week_days[iso_day - 1],
                "is_omega": False
            }
        days_sum += length
    return None
```

Listing 1: GHH Date Conversion Logic

In addition to individual date conversion, the software includes a feature to generate a full annual GHH calendar for any given Gregorian year. This function iterates through the 364-day fixed cycle (plus the Omega week if applicable). Figure 1 illustrates the first quarter of 2026 under the GHH system, demonstrating the fixed monthly starting days and the continuous weekly cycle.

```
##########################################################################################
   GREEK-HANKE-HENRY PERMANENT CALENDAR (GHH) : YEAR 2026
##########################################################################################
```

ALPHA (30 Days)

Mon	Tue	Wed	Thu	Fri	Sat	Sun
01 (Dec 29)	02 (Dec 30)	03 (Dec 31)	04 (Jan 01)	05 (Jan 02)	06 (Jan 03)	07 (Jan 04)
08 (Jan 05)	09 (Jan 06)	10 (Jan 07)	11 (Jan 08)	12 (Jan 09)	13 (Jan 10)	14 (Jan 11)
15 (Jan 12)	16 (Jan 13)	17 (Jan 14)	18 (Jan 15)	19 (Jan 16)	20 (Jan 17)	21 (Jan 18)
22 (Jan 19)	23 (Jan 20)	24 (Jan 21)	25 (Jan 22)	26 (Jan 23)	27 (Jan 24)	28 (Jan 25)
29 (Jan 26)	30 (Jan 27)					

BETA (30 Days)

Mon	Tue	Wed	Thu	Fri	Sat	Sun
		01 (Jan 28)	02 (Jan 29)	03 (Jan 30)	04 (Jan 31)	05 (Feb 01)
06 (Feb 02)	07 (Feb 03)	08 (Feb 04)	09 (Feb 05)	10 (Feb 06)	11 (Feb 07)	12 (Feb 08)
13 (Feb 09)	14 (Feb 10)	15 (Feb 11)	16 (Feb 12)	17 (Feb 13)	18 (Feb 14)	19 (Feb 15)
20 (Feb 16)	21 (Feb 17)	22 (Feb 18)	23 (Feb 19)	24 (Feb 20)	25 (Feb 21)	26 (Feb 22)
27 (Feb 23)	28 (Feb 24)	29 (Feb 25)	30 (Feb 26)			

GAMMA (31 Days)

Mon	Tue	Wed	Thu	Fri	Sat	Sun
				01 (Feb 27)	02 (Feb 28)	03 (Mar 01)
04 (Mar 02)	05 (Mar 03)	06 (Mar 04)	07 (Mar 05)	08 (Mar 06)	09 (Mar 07)	10 (Mar 08)
11 (Mar 09)	12 (Mar 10)	13 (Mar 11)	14 (Mar 12)	15 (Mar 13)	16 (Mar 14)	17 (Mar 15)
18 (Mar 16)	19 (Mar 17)	20 (Mar 18)	21 (Mar 19)	22 (Mar 20)	23 (Mar 21)	24 (Mar 22)
25 (Mar 23)	26 (Mar 24)	27 (Mar 25)	28 (Mar 26)	29 (Mar 27)	30 (Mar 28)	31 (Mar 29)

Figure 1: Visual representation of the first quarter of 2026 in the GHH system.

5 Conclusion

The Greek-Hanke-Henry Permanent Calendar represents a logical evolution of the Hanke-Henry proposal. By adopting a scientific Greek nomenclature, it solves the interface and ambiguity problems that plague calendar reform.

AI Declaration

While the majority of the drafting of this article and the Python implementation were primarily performed by Gemini 3 Pro, the author initiated all research directions and verified all content, citations and discussions of existing literature. The author retains full responsibility for the content and conclusions presented herein.

Code Availability

The Python implementation of the Greek-Hanke-Henry Permanent Calendar is available on GitHub at `https://github.com/denniskhong/ghh-calendar`.

References

Bushell, W. F. (1961). Calendar reform. *The Mathematical Gazette*, 45(352), 117–124. https://doi.org/10.2307/3614624

David, P. A. (1985). Clio and the economics of QWERTY. *The American Economic Review*, 75(2), 332–337. http://www.jstor.org/stable/1805621

Davies, C., Trivizas, E., & Wolfe, R. (1999). The failure of calendar reform (1922–1931): Religious minorities, businessmen, scientists, and bureaucrats. *Journal of Historical Sociology*, 12, 251–270. https://doi.org/10.1111/1467-6443.00091

Hanke, S. H., & Henry, R. C. (n.d.). *The Hanke-Henry Permanent Calendar*. http://hankehenryontime.com/html/calendar.html

International Organization for Standardization. (2019). *ISO 8601-1:2019. Date and time—Representations for information interchange*. Geneva, Switzerland: ISO https://www.iso.org/standard/70907.html