# A Precision Internet-Connected Clock Timer

## *Its Application to Improving Wheel Train Performance*

Hazel Mitchell

Mike Godfrey

Alastair Godfrey

### *Introduction*

As part of a larger project to design and manufacture an ultra-precision pendulum regulator, we required a method to measure accurately the rate of the pendulum. Commercial clock timing solutions, for instance MicroSet or Timetrax timers, can be purchased, but are often expensive, insufficiently accurate (unless equipped with GPS) and do not offer internet connectivity[1+2].

Data corruption, jumping between left and right swings, electrical interference and clashing of the reference tick with the pendulum tick are all problems reported with MicroSet timers[3] and each of these issues has been addressed with the present design. Furthermore, an internet-connected device allows data logging and remote data processing, in real time, without requiring a local computer to be connected.

Almost all available clock timing devices use a crystal oscillator as a reference timer, which is not accurate enough for an ultra-precision clock. To put this into perspective, an ultra-precision regulator, such as Burgess 'Clock B', is claimed to be able to keep within a second of deviation over 100 days of operation, i.e. approximately ±0.1 ppm[3]. Most 'real-time clock' (RTC) modules* available are accurate to ±2 ppm (e.g. RV3028), which is over an order of magnitude less stable than the measured timepiece itself.

We felt a reference timer that is ten times more accurate than the timepiece to be measured would suffice, but 100 times more accurate would be preferable, which would be ±1 ppb. It is possible to purchase high-end oscillators that are able to achieve this level of accuracy, for example IQD manufactures an oscillator that is rated to ±1.5 ppb, but these are expensive. Network Time Protocol (NTP) server time, whilst accurately disciplined to GPS or a national time standard over the long-term, does not provide an accurate enough short-term reference for the sorts of measurements we wish to perform, due to delays in network routing.

However, one method of achieving an accurate reference timer is the use of GPS time, which is required to be accurate to within tens of nanoseconds in order to calculate an accurate position. GPS receivers are affordable, and many modules even have a pulse-per-second (PPS) output pin.

Fundamentally, our approach makes use of the PPS pin on a GPS receiver and measures the time between the GPS pulse and the pulse from an optical detector in the form of a light gate to achieve accurate timepiece rate measurement. Furthermore, by measuring the pass duration of a strip attached to the pendulum, the velocity of the pendulum at the lowest point of its swing can be calculated and the pendulum amplitude inferred from this. We also added temperature, barometric pressure and humidity sensors.

We designed the timer to be able to upload data directly to the internet so we could view and analyse the data in real-time remotely, and store data on a MicroSD card as a back-up, should the internet connection fail. We are able to measure the amplitude and rate with extremely high precision and log temperature, humidity and barometric pressure data, with a device costing less than £100.

### *Operating Principle*

The principle of this timer is based on an incredibly reliable GPS pulse received once every second, combined with a light gate triggered by the pendulum at approximately the same frequency. The pendulum used here is a one-second pendulum – i.e. a period of two seconds. Most importantly, even if random fluctuations occur due to the time taken for the timer code to execute, there is no possibility of long-term drift in the reference GPS signal.

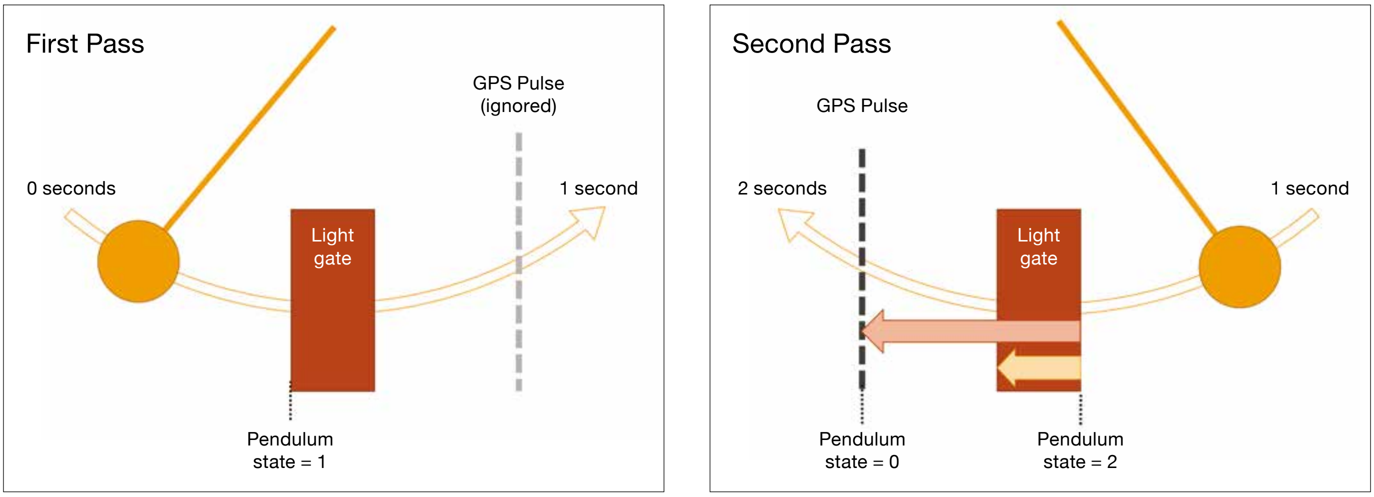
A key aspect to the design of this device is its ability to differentiate between left and right pendulum swings – critical if the light gate is not perfectly positioned in the centre of the pendulum swing, which invariably it won't be.

The same issue has been previously reported with MicroSet timers, even for the measurement of Burgess Clock B.[3+4] Our solution is implemented using 'pendulum states'. On the first pass, the light gate is triggered, setting the pendulum state to one. Whilst the pendulum state is one, the GPS signal is ignored. On the second pass, the light gate is triggered again, setting the pendulum state to two, **Figures 1A and 1B**. At this stage, three time points are recorded:

A. When the pendulum begins passing through the gate.
B. When the pendulum leaves the gate.
C. When the next GPS pulse is received.

After the GPS pulse, the pendulum state is reset to zero, the time data is calculated and the data is transmitted to an auxiliary microcontroller for further processing. The pass

* RTCs are electronic circuits that measure the true time of day, as opposed to oscillators that simply supply reference pulses. – *JJK*




*Figures 1A and 1B. The basic acquisition methodology of the device.*

duration is calculated as the time the pendulum left the gate minus the time it entered (B-A). The phase offset is the time when the GPS pulse was received minus the time the pendulum entered the gate (C-A).

## *Design and Hardware*

The hardware consists of two microcontrollers: an Arduino Nano and an Arduino Mega, with various sensors connected. The only external connections are a USB power supply and an ethernet cable connected to a local router. Functions within the Arduino script known as 'interrupts' are run instantly when a rising or falling edge is detected on the specified input pins. The Arduino Nano handles only the time-sensitive interrupts from the GPS module and light gate, then transmits this data over a serial connection to the Arduino Mega.

All of the processes handled by the Arduino Mega are less urgent and can occur independently from the timer itself. This setup was selected to prevent clashes between the interrupts and time-consuming processes, like the ethernet data transmission or display refreshes. Clashing between the pendulum and reference tick was a reported problem during the measurement of the Burgess Clock B with a MicroSet timer[3]. Another advantage is that the program storage on the Mega is larger than the Nano, facilitating the use of large software libraries, such as a graphics library for the display. An overview of the design is shown in **Figure 2**.

A custom stackable circuit board was fabricated to fit all of the sensors and the Nano on top of the Mega and ethernet shield. It is worth noting that brand new official Arduino boards are much more expensive than the 'clones' available on websites like eBay. These unofficial boards usually offer identical functionality but may require more effort and debugging to render operational. An overview of official component prices at the time of writing is provided in **Table 1**.

The Arduino Nano is a 16 MHz microcontroller, meaning the theoretical maximum precision of the timing functions is 4 µs[5]. Timing errors can occur during the period when the pendulum pulse processing overlaps the GPS pulse processing; however, this is very unlikely and splitting processes between the Nano and Mega reduces the probability of this issue occurring.

Currently, the LCD display shows live phase offset, temperature, pass duration and pressure, updated once every five seconds, but this can easily be adjusted to suit the user's preferences. Meanwhile, every 15 seconds, the most recent

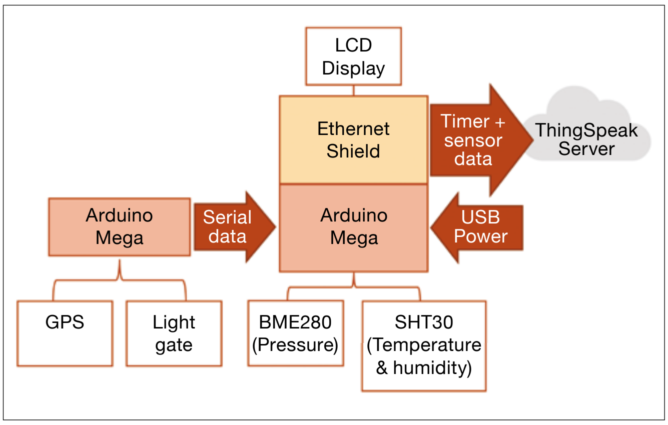



*Figure 2. A block diagram showing the fundamental design of the timer.*

| **Component** | **Function** | **Price** |
|---|---|---|
| Arduino Nano | Timing measurement | £10.80 |
| Arduino Mega | Handles environmental sensors, additional processing and data transmission | £36.00 |
| Ethernet Shield | Upload data to ThingSpeak or save to a MicroSD card | £10.31 |
| Ethernet cable (5m) | Upload data to ThingSpeak | £3.79 |
| Arduino power cable (USB-A to USB-B) | 5V power supply | £2.31 |
| GPS module | 1 Hz PPS reference signal | £6.02 |
| GPS antenna (3m) | Improve GPS reception | £6.99 |
| Light gate | Triggered by pendulum passes | £4.72 |
| BME280 | Measure pressure | £3.77 |
| SHT30 | Measure temperature and humidity | £2.45 |
| 1.3" LCD display | Display live data | £7.00 |
| MicroSD card (optional) | Save data locally | £2.90 |
| | | **£97.06** |

*Table 1. A breakdown of the components and their respective costs.*

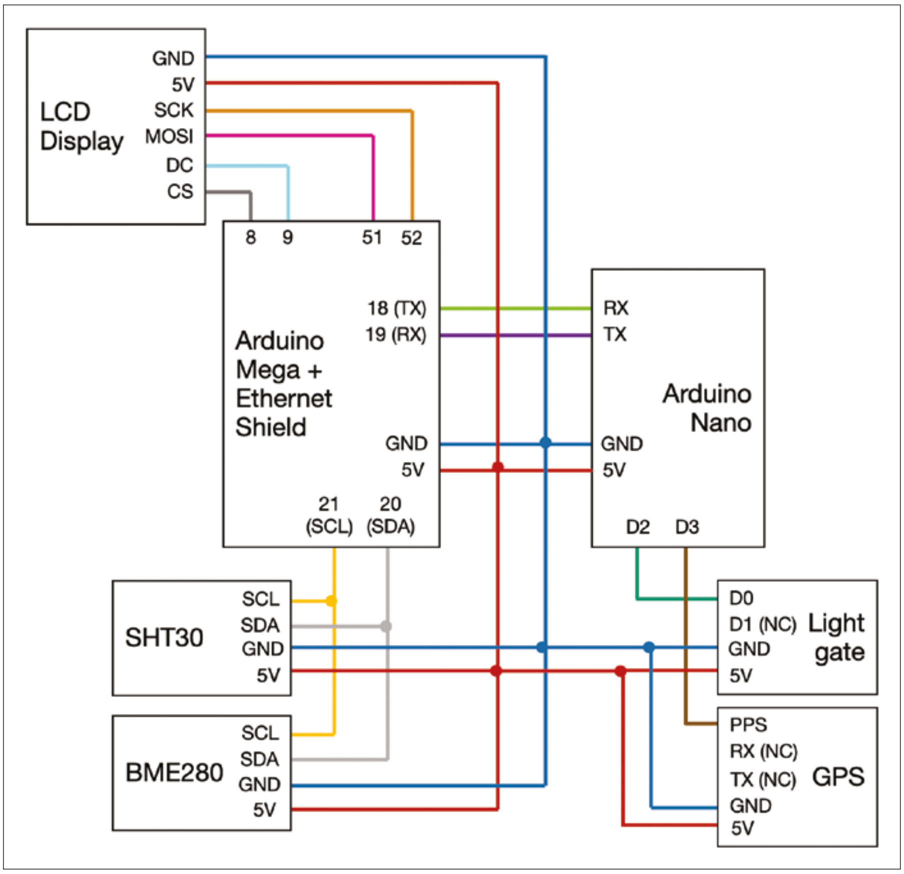



*Figure 3. A circuit diagram of the clock timer.*

data point is uploaded to ThingSpeak, a cloud-based data analysis service, via an ethernet connection.

Sampling once every 15 seconds is an unusually high sampling frequency; however, interesting phenomena can be measured that would otherwise be entirely missed. A free non-commercial ThingSpeak account was used for this system, limiting the upload interval to fifteen seconds, but more frequent uploads can be enabled by purchasing a ThingSpeak subscription. Alternatively, data can be saved directly to a MicroSD card. No averaging is performed on the data locally, to avoid false data being generated by GPS signal loss or other disturbances.

The ThingSpeak website displays live adjustable graphs of the phase offset, pass duration, temperature, humidity and pressure, as well as derived plots, such as dew point or air density vs amplitude. Even more intensive data analysis can easily be performed by downloading all of the recorded data directly through MATLAB, or as a .csv file, for use in the user's preferred software. For those who wish to reproduce the device, a circuit diagram is shown in **Figure 3** and the code is accessible from a GitHub repository[6].

### *Amplitude Measurement*

Without a timing device, accurate amplitude measurement is tedious. One example is Pierre Boucheron's excellent article *Sources of Pendulum Losses*, in which their experimental apparatus was a complex arrangement involving a mirror mounted to the pendulum rod, onto which a narrow light slit was projected, and a movable photodetector mounted on a carriage[7].

The present device does not measure amplitude directly – instead, we designed the device to export the duration the light gate is blocked by the passing strip; the conversion to amplitude is performed in ThingSpeak, where calibration and adjustments are logistically easier.

We chose to use the same light gate for the amplitude measurement as for the rate measurement. This was because at the lowest point in the swing, the pendulum experiences the least acceleration, thus allowing us to assume a constant velocity as the strip passes through the light gate. It may be argued that there is more error because the pendulum is travelling fastest at this point, thereby reducing pulse length. In practice though, we achieved accurate amplitude measurement with a strip 9 mm wide, corroborated with a manual measurement from a graduated scale situated underneath the pendulum. Furthermore, the pulse length can easily be lengthened by increasing the strip width, if required.

Extraction of the amplitude from the pulse length is very simple. The potential energy of the pendulum at the highest point in the swing can be equated with the kinetic energy of the pendulum at the lowest point in the swing (**Equation 1**), rearranged so the subject is the height difference of the pendulum (**Equation 2**) and the height difference is converted to an amplitude using trigonometry (**Equation 3**). The geometrical values used in the equations below are defined in **Figure 4**:

$$\frac{1}{2}mv^2 = mgh \qquad \text{(Equation 1)}$$

$$\Rightarrow h = \frac{v^2}{2g} = \frac{w^2}{2gT_{pulse}^2} \qquad \text{(Equation 2)}$$

$$\varnothing = \cos^{-1}\left(\frac{L-h}{L}\right) \qquad \text{(Equation 3)}$$

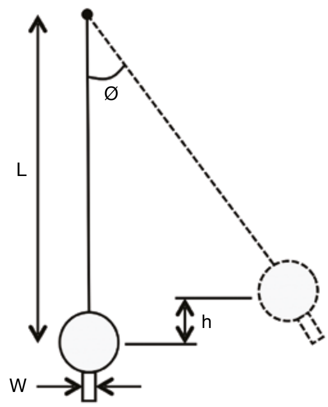



*Figure 4. Definitions of values used in the equations above.*

Thus, the amplitude (ø) can be calculated where the independent variable is the period of time for which the light gate is blocked by the strip ($T_{pulse}$). To calibrate the amplitude measurement, the value for strip width (w), is altered such that the amplitude calculated by the method outlined above agrees with the amplitude measured by another method – in our case, we used a graduated scale located underneath the pendulum. The strip width value should be close to the actual value; however, beam width, light amplitude changes, air resistance and suspension losses are not accounted for in the model and, therefore, the values may deviate slightly. Once calibrated for one value of ø, the model accurately predicts the amplitude across the whole range, as shown in **Figure 5**.

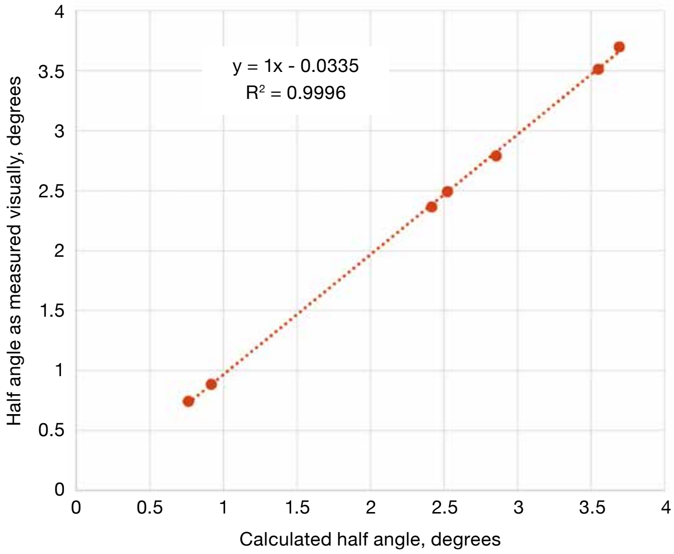



*Figure 5.* **Gate Calibration:** *Amplitude measured with graduated scale vs amplitude calculated by the clock timer.*

## *Application to Improving Drive Train Efficiency and Stability*

In **Figure 6**, the gate time is converted to pendulum amplitude in degrees, as described above; the high level of amplitude resolution is obvious. This simple time series plot of the amplitude clearly shows repetitive patterns in the amplitude of the pendulum swing, which can only be interpreted as variations in the efficiency with which drive power is transferred to the oscillator.

In a pendulum clock, unless special precautions are taken such as the use of a remontoire or a gravity escapement, variations in the drive efficiency unavoidably give rise to variations in rate. This is due mostly to circular deviation stemming from variations in pendulum amplitude. Variation in rate is typically the thing we are most anxious to minimise in our quest for a precision regulator. The largest variations of amplitude on this plot have periods of one hour and 1/8 th of an hour.

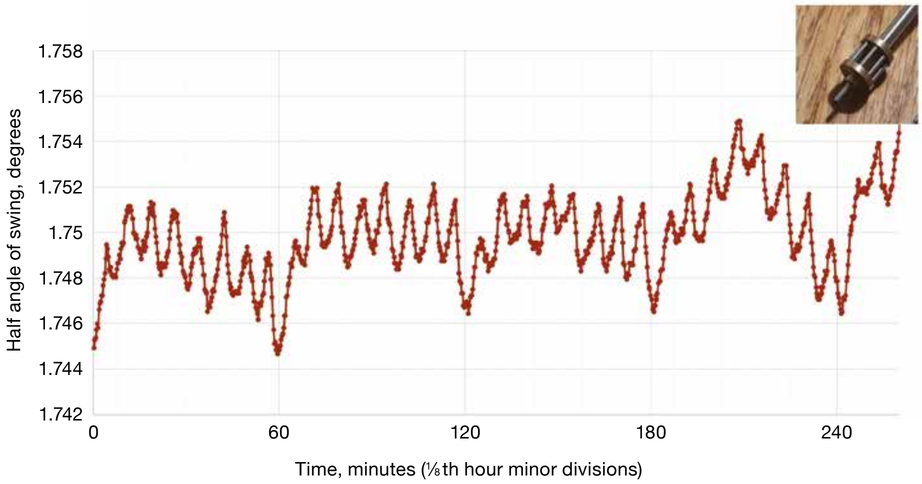



*Figure 6.* **Pendulum Amplitude:** *Regular variations in pendulum amplitude due to variability in wheel train efficiency during each individual tooth mesh with a lantern pinion (top right of figure) on the minute mobile.*

These periodicities were characteristic of the wheel train of the clock under investigation and could be traced back to an imbalance of the minute wheel, and a non-optimal pinion leaf profile – an eight-leaf lantern pinion on the minute wheel arbor. These were then mostly rectified by replacing the pinion and balancing the minute wheel, see **Figure 7**. Also note the increased amplitude of the pendulum with the replacement pinion, indicating increased drive efficiency.

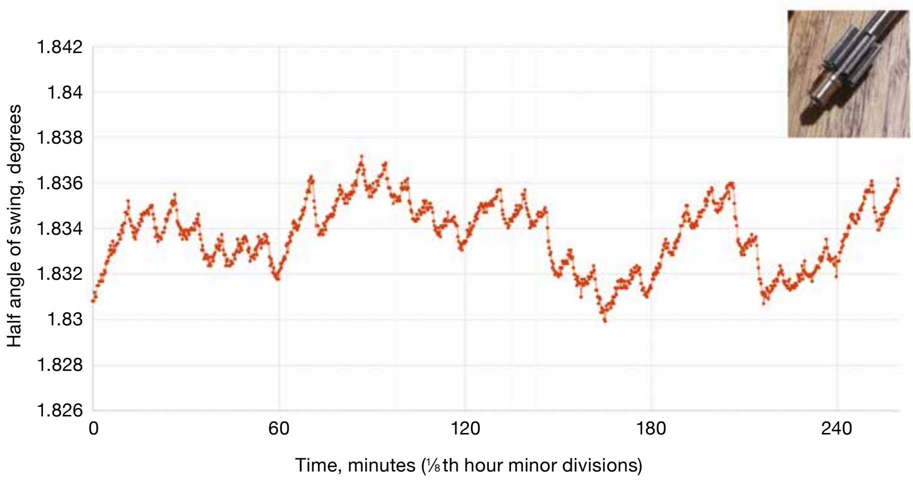



*Figure 7.* **Pendulum Amplitude, New Pinions, Balanced:** *A reduction in amplitude variation and an overall increase in amplitude with a machined pinion (top right of figure) on a poised minute mobile.*

The availability of this high-resolution data enables the detailed investigation needed to rectify all possible factors contributing to variation in amplitude. A Fourier analysis of a longer dataset can be used for a full breakdown of the components that give rise to variation.

During testing, there were a few observations that are worth noting.

We found that official Arduino components were more reliable than third party copies. Even with official Arduino components, very occasionally the internet connection was temporarily lost; this may be due to network instability. However, with the SD card installed, data is never lost during an important experiment. We never observed any electrical interference problems and although theoretically possible, we haven't experienced clashing between the PPS pulse and the clock tick with the dual microcontroller approach.

### *Conclusions*

In this article, we presented a precision internet-connected clock timer that rivals commercially available solutions. We demonstrated that, with its high sensitivity combined with frequent sampling, individual tooth force transmission variability was measurable in the amplitude data of a precision regulator. The device is open-source under a CC BY-NC-SA 4.0 license, and we encourage experimentalists to employ the design in their own projects.


### *About the Authors*

Hazel Mitchell, Mike Godfrey and Alastair Godfrey are engineers at Chronova Engineering Ltd, <https://chronova-engineering.square.site>



REFERENCES

1. Timetrax Model 185 Timer, 2022 <https://www.cousinsuk.com/product/timetrax-model-185-beat-amplifier-escapement-tester> (Accessed 21 October 2022).
2. MicroSet 3 Timer, 2022 <https://www.bmumford.com/mset/microset-3/> (Accessed 21 October 2022).
3. McEvoy, R. and Betts, J., *Harrison Decoded: Towards a Perfect Pendulum Clock* (Oxford University Press, 2020).
4. McKay, C., 'Timing Turret Clocks and Diagnosing Faults 2', *Horological Journal*, vol. 154 (April 2012), p172.
5. Arduino Micros Function, 2022 <https://www.arduino.cc/reference/en/language/functions/time/micros/> (Accessed 21 October 2022).
6. Mitchell, H., GitHub Repository, 2022 <https://github.com/ChronovaEngineering/Arduino_Clock_Timer> (Accessed 21 October 2022).
7. Boucheron, P., 'Sources of Pendulum Losses', *Horological Journal*, vol. 133 (March 1991), p312.