Hollow Clock

# Introduction

## Hollow Clock V

This clock is based on the plans in Thingiverse. It is sold under license from the designer. I have, however, modified the software. The software now has a serial menu system that can be used to fine tune the clock’s accuracy and several other settings. I also changed the software to use the µSecond (microsecond) counter for more accuracy. The original code used the µSecond counter multiplied by 1000 to give µSeconds, but this of course doesn’t really increase the accuracy since the counter only reads mSeconds. In addition to this the code did not handle the counter overflow that happens about every 49 days. This caused a gain of up to a minute each time the overflow happened.

# Serial Port Menu System

## USB Connection to Computer

The clock must be connected to the computer to use the menu system. A USB C to A or C type connector may be used. The PC must be running some serial port software, such as SSCOM or similar. The Arduino IDE is an excellent choice, it is free and easy to install. Enable the serial port monitor and a section of the IDE will display the text coming from the clock.

## Received Data

Every minute a line will be sent from the clock. This information is useful for calibrating the clock.

Here is a sample line.

20:52:14.172 -> minutes: 196 Hours: 3.27 current uS: 3230069237 last uS: 3170065408

Note that the time tag at the start does not come from the clock, it is a tag displayed by the terminal software and is very useful for calibrating the clock speed. The Hours field shows how long the clock has been running. The last two numbers are just the values of the µSecond timer used by the clock. These numbers were mostly used for debugging but have been left in the line in case anybody wants to see them.

## The Menu

The menu is displayed after entering ‘?’ or simply by entering a blank line in the serial terminal software. Each command is a single character, some of which require additional information after the command.

20:53:20.343 -> ----- Current Settings -----

20:53:20.343 -> Data version : 1

20:53:20.343 -> uSeconds adjust per minute : 0

20:53:20.343 -> Reverse Motor : No

20:53:20.343 -> Test Mode : Off

20:53:20.343 -> Stepper Delay : 6

20:53:20.343 -> Wait State : Running

20:53:20.343 -> ----- Commands -----

20:53:20.343 -> +<n> : Advance n minutes

20:53:20.343 -> -<n> : Reverse n minutes

20:53:20.343 -> A<n> : Adjust Minute Position (+/- 256 is a full minute)

20:53:20.343 -> T : Test mode (enter anything while running to stop)

20:53:20.343 -> S<n> : Set stepper motor delay, default is 6, range 2 to 120

20:53:20.343 -> C<n> : Calibrate uSeconds per minute, is default, change as needed, +speeds up, -slows down

20:53:20.343 -> F<sec> <hours> : Figure correction using seconds and hours (floats), e.g. F -2.5 24.0 if 2 seconds slow per day

20:53:20.343 -> R : Reverse motor setting

20:53:20.343 -> W : Wait, toggle running state of clock

20:53:20.343 -> Command?

The first part of the display shows some current values. The last part lists the available commands with short descriptions.

### Commands

#### +<n>

The integer <n> is used to advance the clock by that many minutes. E.g. +5 will move the clock forward by 5 minutes.

#### -<n>

The integer <n> will move the clock backwards by that many minutes.

#### A<n>

This value is used to adjust the minute pointer by small increments, so it lines up with the minute marks exactly. Both positive and negative values may be entered. If a negative value is entered the clock will move backwards and forwards again to take up the gear drive slack for more accuracy.

#### T

This command toggles test mode. When test mode is on the clock runs continuously. This can test the servo and gears to make sure it rotates freely. Pressing any key and enter will stop the test mode and return to normal running.

#### S<n>

S is used to set the stepper motor stepping delay. Weaker motors might need longer delays, or you might just like the different sound of slower or faster stepping.

#### C<n>

This command is used to adjust the running speed of the clock in µSeconds per minute. Positive values make the clock run faster and negative values make it run slower. This value can be calculated from monitoring the clock run for several days. Note that the F command is easier to use and will calculate the correct value to use.