



MAKER: Gyro'clock - The spinnable time reader

Kasun Sanjaya Somaratne, British Columbia Institute of Technology

Kasun Somaratne is a second year Electrical and Computer Engineering Technology student at the British Columbia Institute of Technology (BCIT). He explores his interest in electronics and creativity through innovative projects and experiments. His design for a wireless signal indicator vest for cyclists won the 2014 BCIT Student Innovation Challenge Award in the applied research category. He aspires to become an electronics engineer to help advance the field of electronics and to pursue his passion in creating innovative electronic gadgets.

Gyro'clock – The spinnable time reader

Abstract

Gyro'clock is a portable time keeping device that displays time when spun in the air by hand. The displayed image is synchronized with the rotational frequency to appear at the same position in space by measuring and analyzing the centrifugal acceleration experienced by the device. This project is a fun way to demonstrate the forces acting on a spinning object, and how they can be utilized to determine its rotational frequency. A time set feature is implemented by using an accelerometer to recognize gestures.

Introduction

The so-called "Persistence of vision" (POV) is the idea that our ability to interpret visual signals has a finite speed. This property of an observer can be utilized to show a two-dimensional image with a series of one-dimensional images. POV displays are ever so popular among electronics hobbyists and have found uses in bicycle wheel displays, wall clocks and visual arts. A two-dimensional POV image is created by moving a row of LEDs rapidly across a space while drawing the image one row of pixels at a time. If the image is drawn faster than the time an afterimage "persists" on the human eye (approximately one twenty-fifth of a second), then the image is perceived as a whole by a viewer¹.

Typical POV displays include a propeller, usually an electric motor, and a trigger mechanism to cause the image to appear at the same position. The trigger mechanism usually involves a Hall effect sensor and a stationary magnet, where the Hall effect sensor output changes when it is in close range with the magnet. Because POV displays require a propeller and a stationary reference for triggering, it is difficult to create small portable POV displays.

My objective for this project is to create a Gyro'clock, which is a portable POV display device that shows time when spun in the air by hand. A similar device, called "Spin", has been constructed by Jussi Angeseleva and Stephen Hughes². My project is an attempt to recreate their work using easy to find components that can be assembled at home. The following sections explain the design challenges I faced and the construction of my first prototype of the Gyro'clock.

Design challenges

One of the major challenges of this project is displaying the image at the same angular position in every cycle. This requires a method to detect when the device is in a specific position along its circular path. The triggering mechanism using a Hall effect sensor and a stationary magnet, like those used on traditional POV displays, cannot be used on the Gyro'clock as it is spun by hand, which produces a rather irregular spin pattern.

Fortunately, one reference vector is available and easily accessible anywhere on Earth: gravity. The acceleration due to gravitational force can be detected using an accelerometer, which is an electronic device that measures acceleration along a particular axis. However, other forces at play on a spinning object also register on an accelerometer. The two most significant forces that act on a spinning object are the centripetal force and gravitational force.

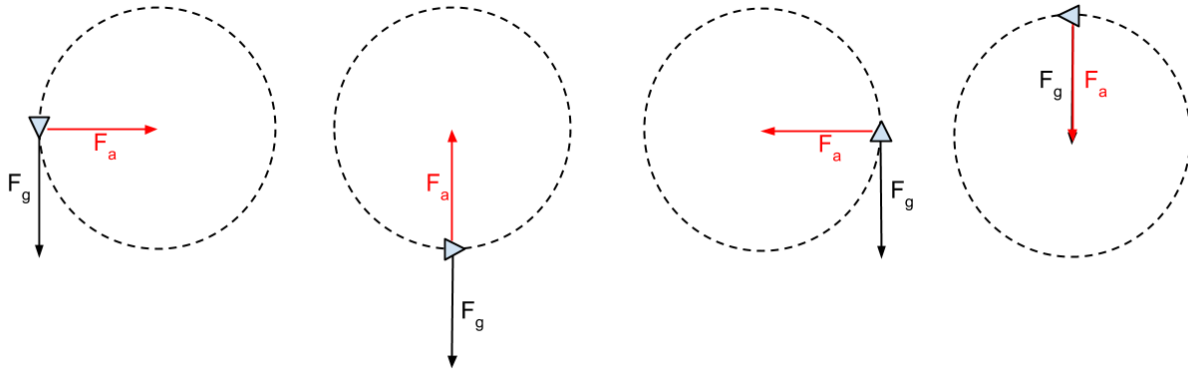


Figure 1: The forces acting on a spinning object at various positions along its circular path, where F_g is the force of gravity and F_a is the centripetal force.

Centripetal force acts towards the center of rotation and makes the device follow a circular path (spin circle). Figure 1 shows the direction of forces acting on an object moving on a circular path perpendicular to the ground. As the object moves from the bottom of the circle to the top, gravity acts against the object's motion, slowing it down. When the object moves from the top of the circle to the bottom, it accelerates due to gravity, achieving a maximum speed at the bottom of the circle. Since the centripetal force is proportional to the square of the velocity, the maximum acceleration due to centripetal force occurs when the device is at the bottom of the circle. Similarly, the minimum centripetal acceleration occurs when the device is at the top of the circle. Thus the centripetal acceleration of a spinning object (under gravity) follows a sinusoidal pattern.

In the frame of the spinning object, the centripetal force is seen as a force that pushes the object away from the center. This apparent force, experienced only by the rotating object, is called the centrifugal force³. An accelerometer attached to a spinning object will measure the centrifugal acceleration as shown in Figure 2. For an object that moves in a circular path, the centripetal force and the centrifugal force are equal in magnitude but opposite in direction. In Figure 2, the accelerometer axis is parallel to the spin circle pointing away from the center. The period of the sinusoidal pattern is equal to the rotation period of the spinning object. By detecting the minimum peaks of the sinusoidal acceleration profile, the image can be triggered to appear at the top of the spin circle. The challenge here is accurately detecting the minima in real time while avoiding false minima created by noise and random jolts experienced by the spinning object.

Another design challenge of this project is making the displayed image always visible to the person spinning the device. When the device is spinning, it tends to rotate around the axis (tumble) connecting the device to the center of the spin circle as shown in Figure 3. Tumbling

can create significant variations in the brightness of the displayed image depending on which way the LEDs are pointing.

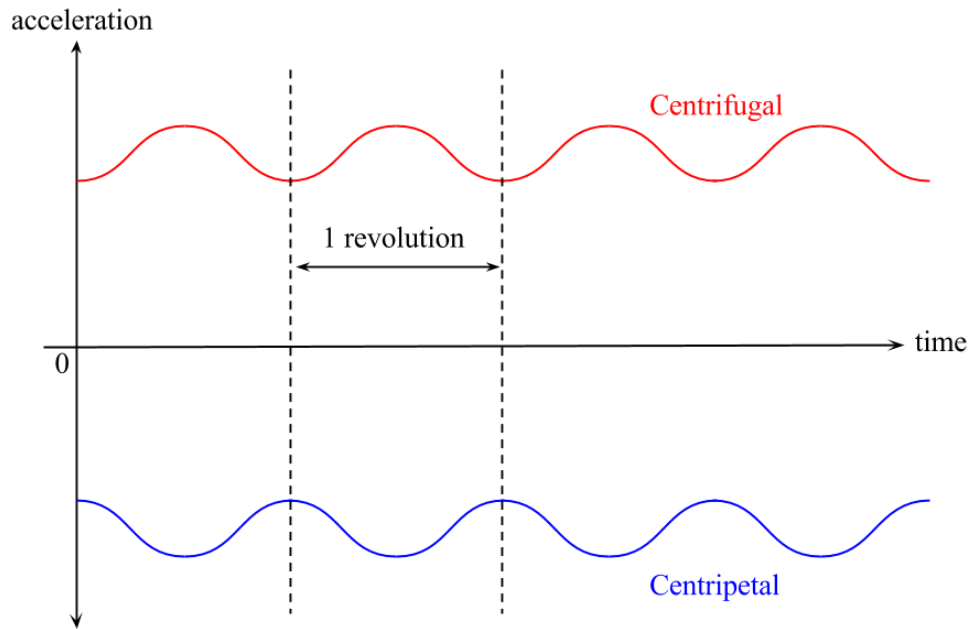


Figure 2: An accelerometer attached to a spinning Gyro'clock will measure centrifugal acceleration. Accelerometer axis is pointing away from the center of the spin circle.

An ideal solution to tumbling is a light pipe that distributes light from an LED uniformly at 360° . Since such light pipes are not readily available, I decided to use a regular LED with a large viewing angle. Viewing angle is the angle at which the brightness of a LED drops to half its maximum value⁴. Viewing angle can range from about 16° to 120° depending on the type of LED and the amount of diffusion created by the cap of the LED⁴. To increase the diffusion of the LED caps, I lightly sanded them with sandpaper.

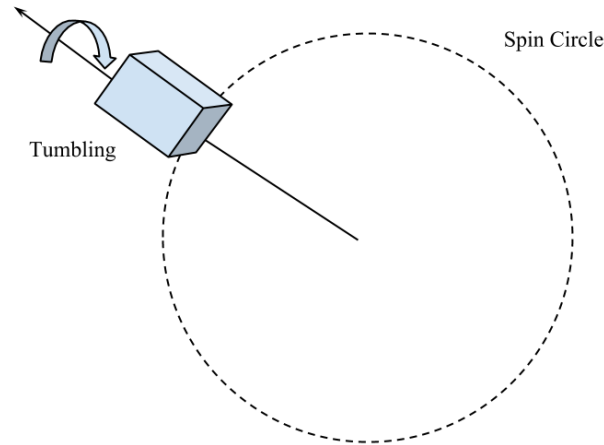


Figure 3: When the Gyro'clock is spinning it tends to rotate around an axis connecting the device to the center of the spin circle (tumbling).

Gyro'clock Schematic

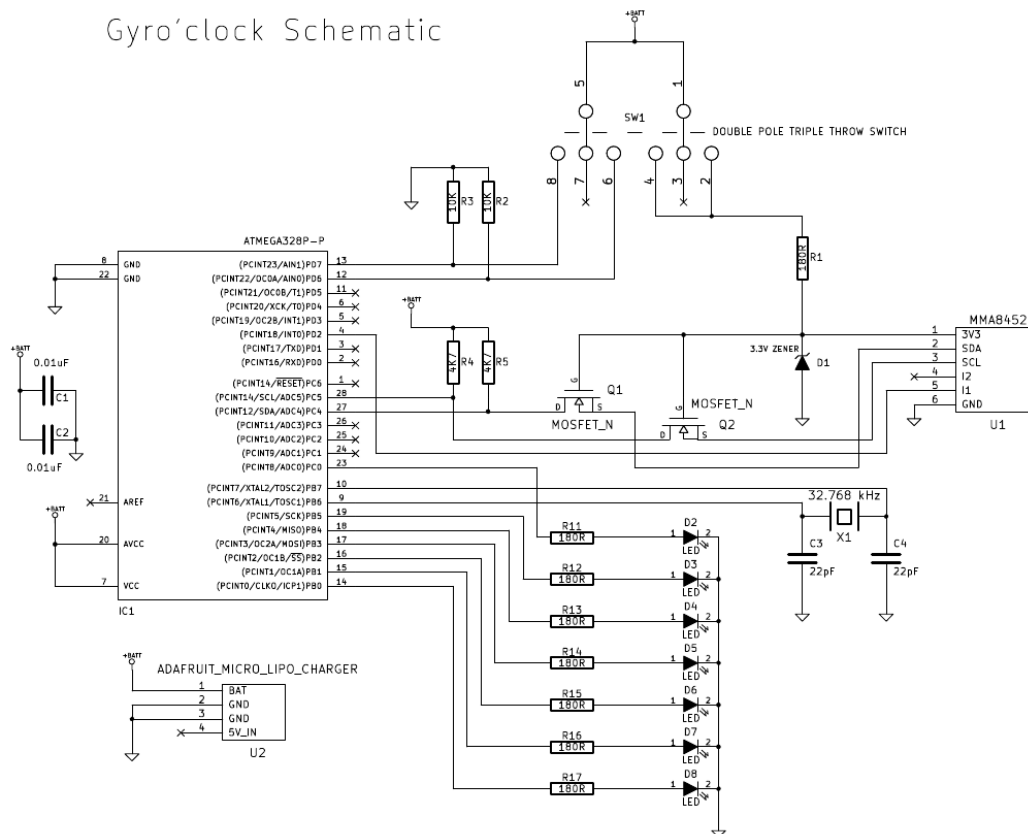


Figure 4: Schematic of Gyro'clock

Circuit design and assembly

The schematic of the Gyro'clock circuit is shown in Figure 4. The circuit is powered by a 150 mAh rechargeable Lithium polymer battery (Lipo battery). The battery voltage varies from 4.2 V (fully charged) to 3.7 V (nominal voltage). However, the MMA8452Q accelerometer is rated for a maximum of 3.6 V. Hence a 3.3 V Zener diode is used to regulate the voltage supplied to the MMA8452Q.

An I²C bus is used to communicate between the accelerometer and the ATmega328p microcontroller. Two n-channel MOSFETs create a bi-directional level shift for the I²C bus, which provides protection for the accelerometer from the high output voltage of the microcontroller⁵.

Table 1: List of materials used for one Gyro'clock

Item	Quantity	Designator
0.01 uF	2	C1, C2
22 pF	2	C3, C4
180 Ω	8	R1, R11 – R17
4.7 k Ω	2	R4, R5
10 k Ω	2	R2, R3
3.3 V Zener (1N4728A)	1	D1
3 mm clear red LED	7	D2 – D8
32.768 kHz crystal	1	X1
n-channel MOSFET (BS170)	2	Q1, Q2
ATmega328p	1	IC1
MMA8452Q – Triple axis accelerometer breakout	1	U1
Double pole triple throw slide switch (DP3T)	1	SW1
28-pin IC socket	1	-
1X10 male header pins	1	-
150 mAh rechargeable Lithium polymer battery	1	-
Adafruit Micro USB Lipo charger (ID:1904)	1	U2
5X8 cm perfboard	1	-
Breadboard wire	-	-
Keychain with ring	1	-

Table 1 lists the materials needed to build a fully functional Gyro'clock. The total cost of the parts was \$36. A fully assembled Gyro'clock is shown in Figure 5. The accelerometer must be mounted onto the board such that one of its three axes is pointing along the radius of the spin circle. The LEDs are placed at the edge of the board protruding outwards to increase visibility. I

decided to not put the board in a case since a case would further restrict the visibility of the LEDs.

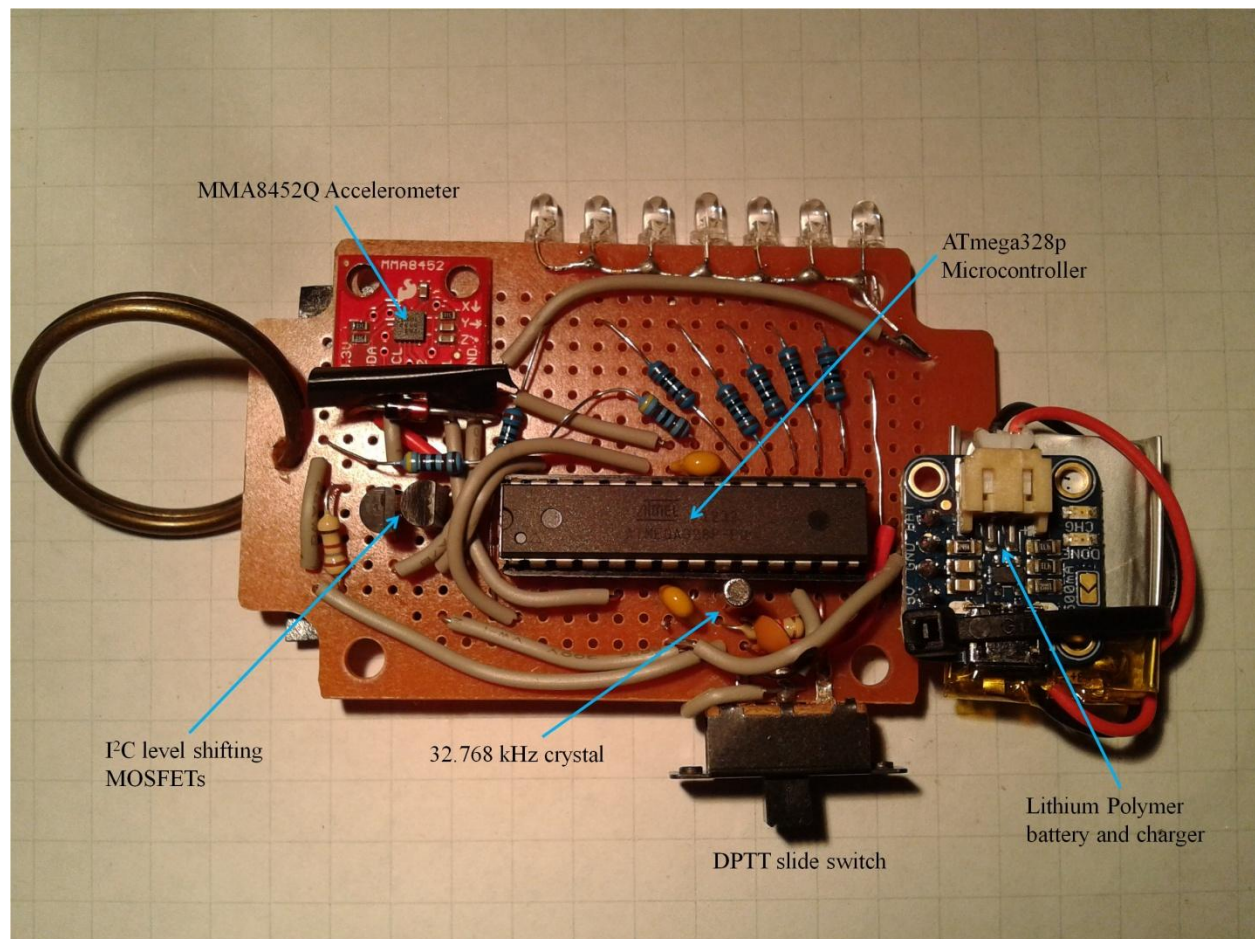


Figure 5: A fully assembled Gyro'clock

Using the Gyro'clock

Gyro'clock has three modes, which can be selected using a slide switch. The three modes are the idle mode, the display mode and the time set mode. In the idle mode only the microcontroller has power for the purpose of keeping time, and the device consumes very low current. With a current consumption of only 155 μA in the idle mode, the device will accurately keep time for 40 days with a fully charged battery.

To view the time, the device must be put in display mode, which will activate power to the accelerometer. Then the device is spun counter-clockwise on a circular path perpendicular to the ground, and the time will appear near the top of the spin circle.

The time set mode allows the user to set the time using gestures. To set the time, the device must be held with the board parallel to the ground. When the device is first put in time set mode, a binary counter is activated, and the count value is displayed in binary using the row of LEDs.

Time is recorded by flipping the device with the LEDs pointing downward when the count value reaches the number of hours and minutes successively. All seven LEDs will turn on for half a second when the count value is recorded.

The rechargeable battery is charged by connecting the Lipo charger using a micro USB cable to a 5V source. A green LED on the charger will indicate when the battery is fully charged.

Finding the trigger point

I decided to make the device show the time at the top of the spin circle, where it appears at the eye level of the spinner. This is also where the device experiences the minimum acceleration (trigger point). To confirm the theory described above and to formulate a reliable trigger point detection algorithm, it was necessary to analyze the output data of the accelerometer. To see the real time accelerometer data of a spinning Gyro'clock, I put together a separate test unit with a Bluetooth module to send data wirelessly to a computer (see Figure 6).

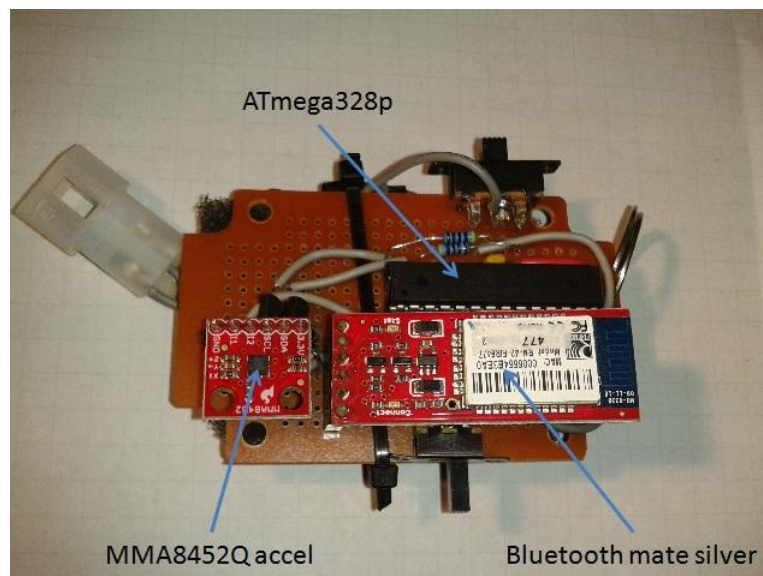


Figure 6: Test unit used to send accelerometer data wirelessly to a computer for analysis

The data shows a roughly sinusoidal profile for the centrifugal acceleration at slower rotational frequencies (< 2 Hz). Although the acceleration profile remains periodic at higher frequencies, the sinusoidal waveform gets clipped by the limited 8g range of the accelerometer (see Figure 7). Increasing the speed further shifts the centrifugal acceleration beyond the range of the accelerometer.

The trigger points correspond to the local minima of the observed acceleration profile. A simple minima detection algorithm will check for points where the data changes from a negative slope to a positive slope. This method will work if the observed accelerometer data has the ideal sinusoidal form. However, the level of noise I observed in the accelerometer output makes such an algorithm ineffective.

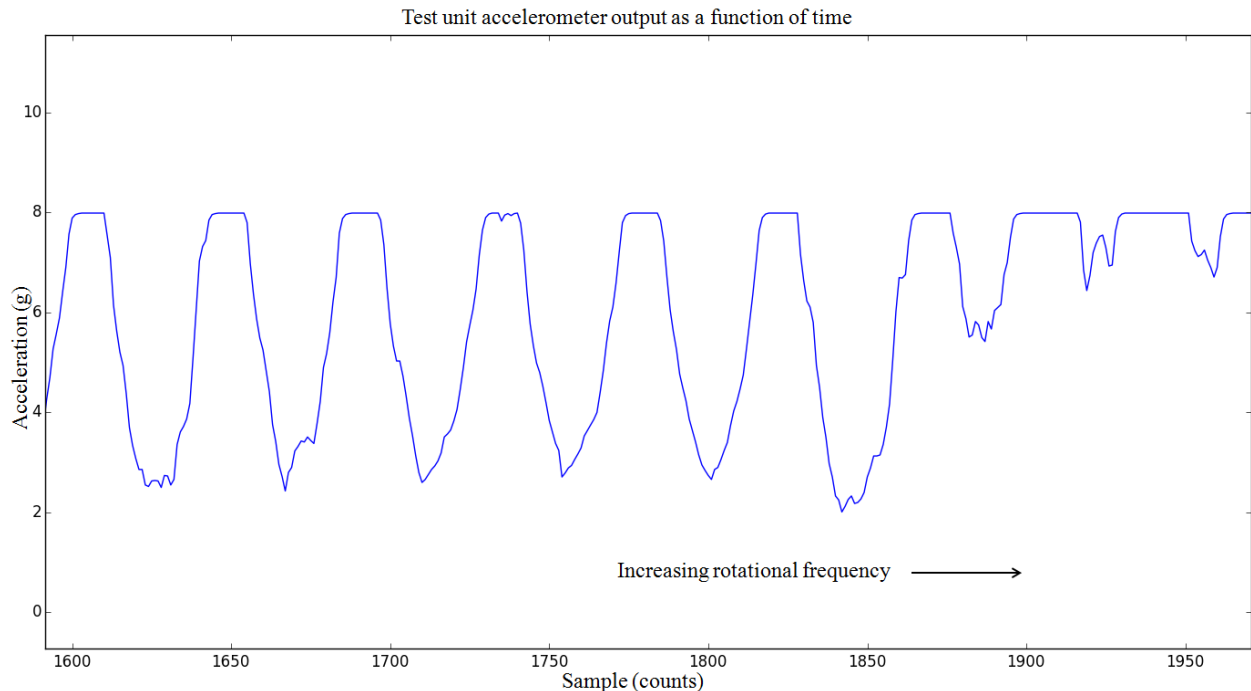


Figure 7: The output of the test unit accelerometer showing a roughly sinusoidal centrifugal acceleration profile. Data is sampled at a 10 ms interval

Due to the complexity of accurately detecting minima of the measured acceleration in real time, it was necessary to establish a set of requirements for an acceptable trigger point detection method. By taking into consideration, the observed noise level in the data at various conditions, the computational ability of the 8-bit ATmega328p microcontroller, and the readability of the displayed time, I have set two requirements for an acceptable trigger detection method. The first requirement is that there should only be a single trigger point for each cycle, where a cycle starts at the top of the spin circle and ends at the top of the spin circle after one revolution.

The second requirement is that the trigger points must occur within $\pm 20^\circ$ of the top of the spin circle when spun at a moderate frequency (~ 2 Hz). This requirement allows a margin of error for the trigger detection algorithm. Noise in the observed accelerometer data may cause a trigger point to be detected slightly ahead or after the real minimum. Also since data from the accelerometer is read at discrete intervals, the trigger points will shift with changing rotational frequency.

In order to limit the number of false trigger points detected, a pre-condition must be established. The device will only check for a minimum once the acceleration has dropped by a certain threshold after a set number (de-bounce) of readings. For a given cycle, the device keeps track of the maximum acceleration value it reads, and waits until the acceleration drops by at least 1g (threshold) after at least 6 readings from the maximum. Once the pre-condition is established, the device accepts the first incoming data point that is greater than or equal to the previous data point as the trigger point.

The threshold level ensures that checking for a minimum is done only when the device is producing a periodically varying acceleration pattern with amplitude of at least the threshold value. The de-bounce counter reduces susceptibility to noise. Increasing both the threshold and the de-bounce counter has the effect of decreasing the maximum rotational frequency, where trigger points can be detected.

Another factor that affects accuracy of the trigger point detection is the sample rate. A higher sample rate provides more data points per cycle, which increases resolution. However, a higher sample rate also increases the number of obstacles for the trigger detection algorithm to overcome, which decreases accuracy.

Through experimentation with different values for the threshold, de-bounce counter and the sample rate, I found that a threshold value of 1g with a de-bounce counter of 6 at a sample period of 6 ms produced acceptable results. The effectiveness of the trigger point detection algorithm at typical rotational frequencies and its limitations at higher rotational frequencies are shown in Figure 8.

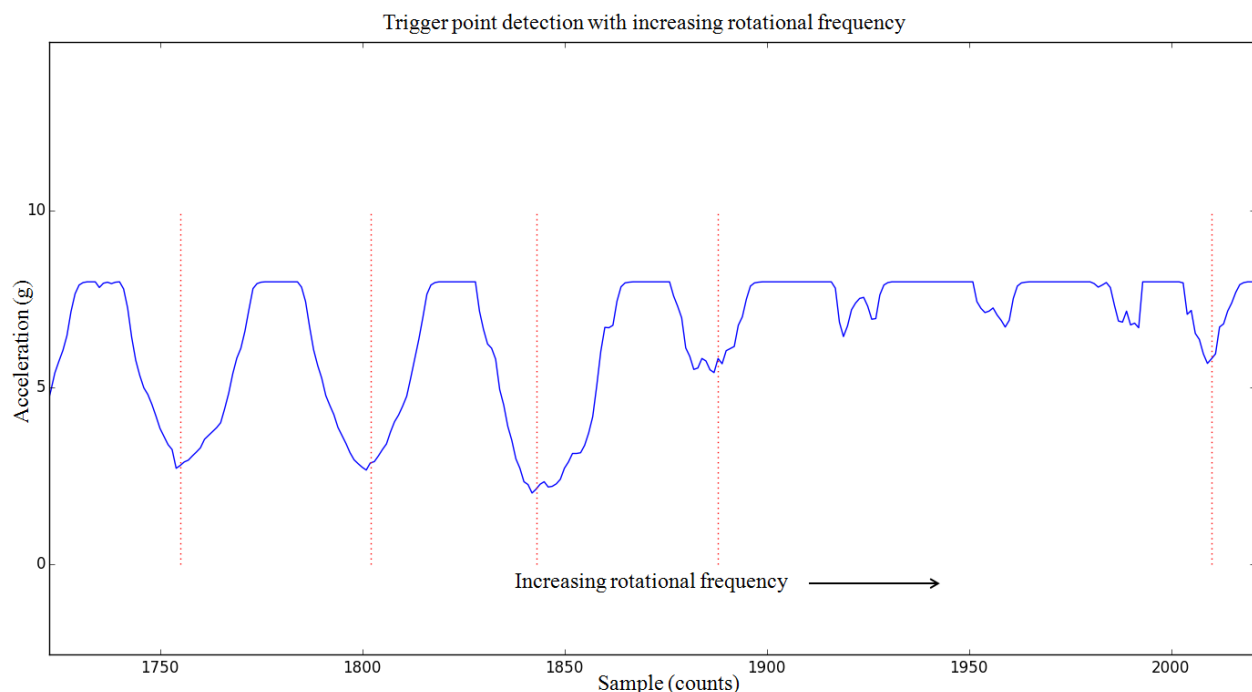


Figure 8: Trigger points (red dotted lines) detected by the minima detection algorithm.

Firmware implementation

The Gyro'clock firmware has four modules: the trigger point detection module, the clock display module, the real time clock module, and the time set module. The centerpiece of all four modules is an ATmega328p microcontroller. This is the same microcontroller that is found on the popular Arduino board. An ATmega328p on an Arduino board runs at a clock speed of 16 MHz, which requires an external crystal. To have a standalone ATmega328p it must be configured to use one

of its internal RC oscillators as the main clock source. Gyro'clock uses the calibrated 8 MHz RC oscillator as its main clock source. The clock source can be configured by flashing an appropriate bootloader to the ATmega328p using another Arduino board as an in-system programmer⁶.

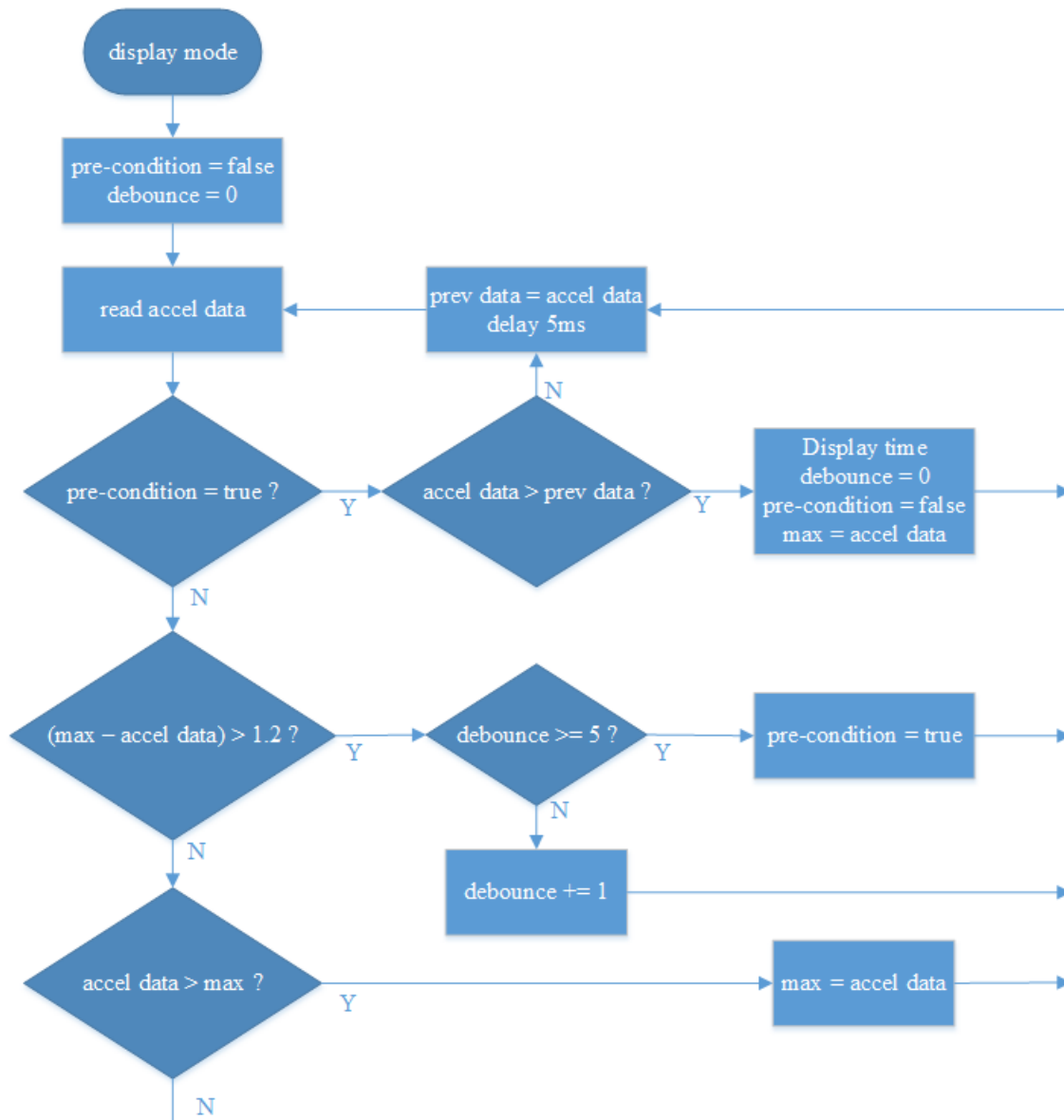


Figure 9: Flow chart for the trigger detection module

The trigger point detection module determines when the Gyro'clock displays time. Digital acceleration data from the MMA8452Q is read every 6 ms using I²C protocol. The noise in accelerometer data is reduced by implementing a leaky integrator type low-pass filter: current data = (0.75 × new data) + (0.25 × previous data)⁷. Figure 9 shows a flow chart for the trigger point detection module.

The clock display module consists of seven LEDs arranged in a row with each LED driven by a microcontroller output. The LED states corresponding to each digit (0-9) are stored as an array of bytes (8-bits). The display sequence for each digit is five bytes in length (see Figure 10). The seven least significant bits of each byte represents the on/off state of the seven LEDs, where a '1' means LED on, and a '0' means LED off. Time is displayed as HH:MM:SS in 24-hour format. Each vertical row of pixels is displayed for 0.5 ms, and the digits are separated by 2 ms.

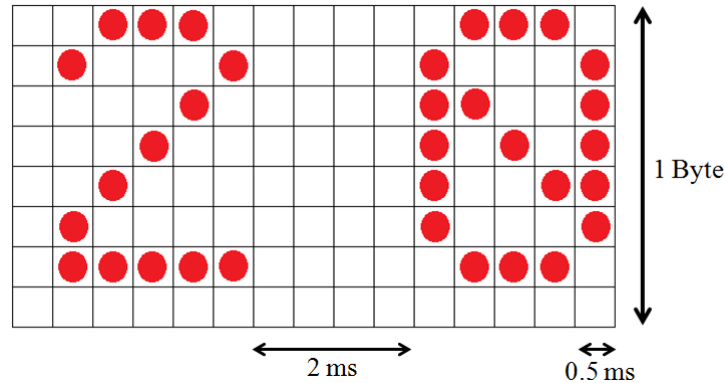


Figure 10: The LED states representing each digit are stored as an array of bytes

ATmega328p has a built-in real time clock module, which keeps track of time accurately using an external 32.768 kHz quartz crystal. The real time clock module utilizes the asynchronous Timer2 counter of the ATmega328p. The 32.768 kHz clock signal is pre-scaled to 256 Hz, which causes the 8-bit counter to overflow in exactly one second⁸.

Using an external asynchronous clock source for time keeping allows the microcontroller to enter a sleep state when in idle mode. In idle mode, the ATmega328p enters the “power-save” sleep mode, which reduces current consumption to just 155 uA. To maximize power savings in the idle mode, I disabled the analog to digital converter, analog comparator, watchdog timer, and the brownout detector of the ATmega328p.

Each clock pulse from the pre-scaled 256 Hz oscillator increments the Timer2 8-bit counter by one. When the Timer2 counter overflows in exactly one second, the overflow interrupt flag is raised, which brings the microcontroller to its active state. The microcontroller enters the Timer2 overflow interrupt service routine and increments the number of seconds by one. Figure 11 shows a flow chart for the Timer2 overflow interrupt service routine. After updating the time, the microcontroller enters the sleep state again, and this process repeats.

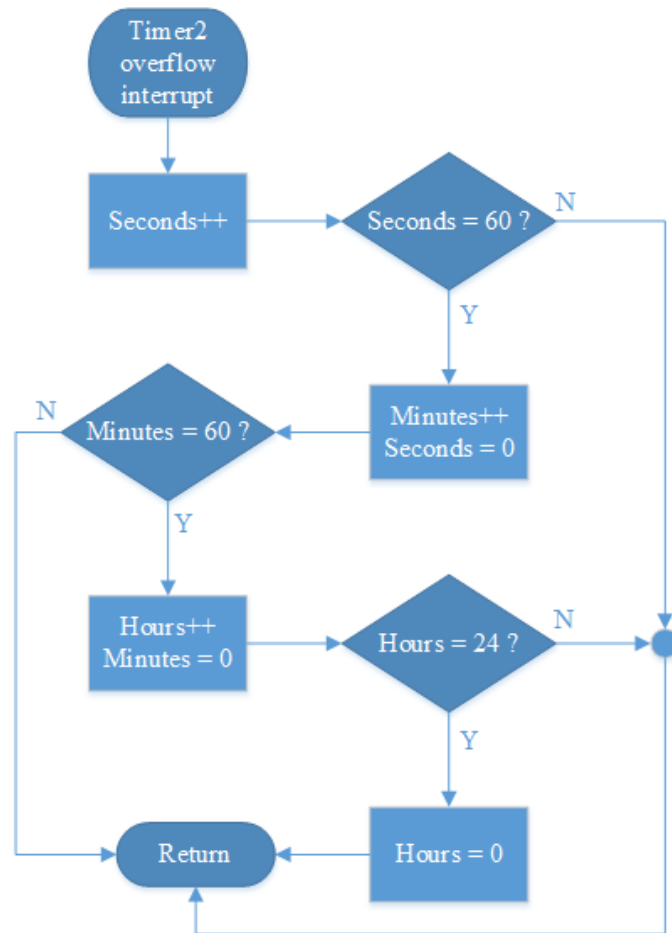


Figure 11: Flow chart for the Timer2 overflow interrupt service routine

The time set module allows the user to set the time using gestures. Figure 12 shows a flow chart for the time set module. In the time set mode, the device recognizes simple hand gestures using the accelerometer readings. This eliminates the need to have additional buttons for setting the time. The device recognizes two gestures: the board is held parallel to ground and the board is held perpendicular to ground with the LEDs pointing at the ground.

In time set mode, when the board is held parallel to ground a binary up counter is activated, which increments every half a second. The counter value is displayed using the LEDs in binary format. When the counter value reaches the number of hours, the user flips the board such that it is perpendicular to ground. When the device is perpendicular to ground, the count value is recorded into the hour variable. All LEDs will turn on for one second when the count value is recorded. After setting the hours the user sets the minutes following the same pattern. Once the hours and the minutes are successfully set, the device enters sleep mode.

The complete source code for the Gyro'clock is available at <https://bitbucket.org/LightEmittingDude/gyroclock/src>.

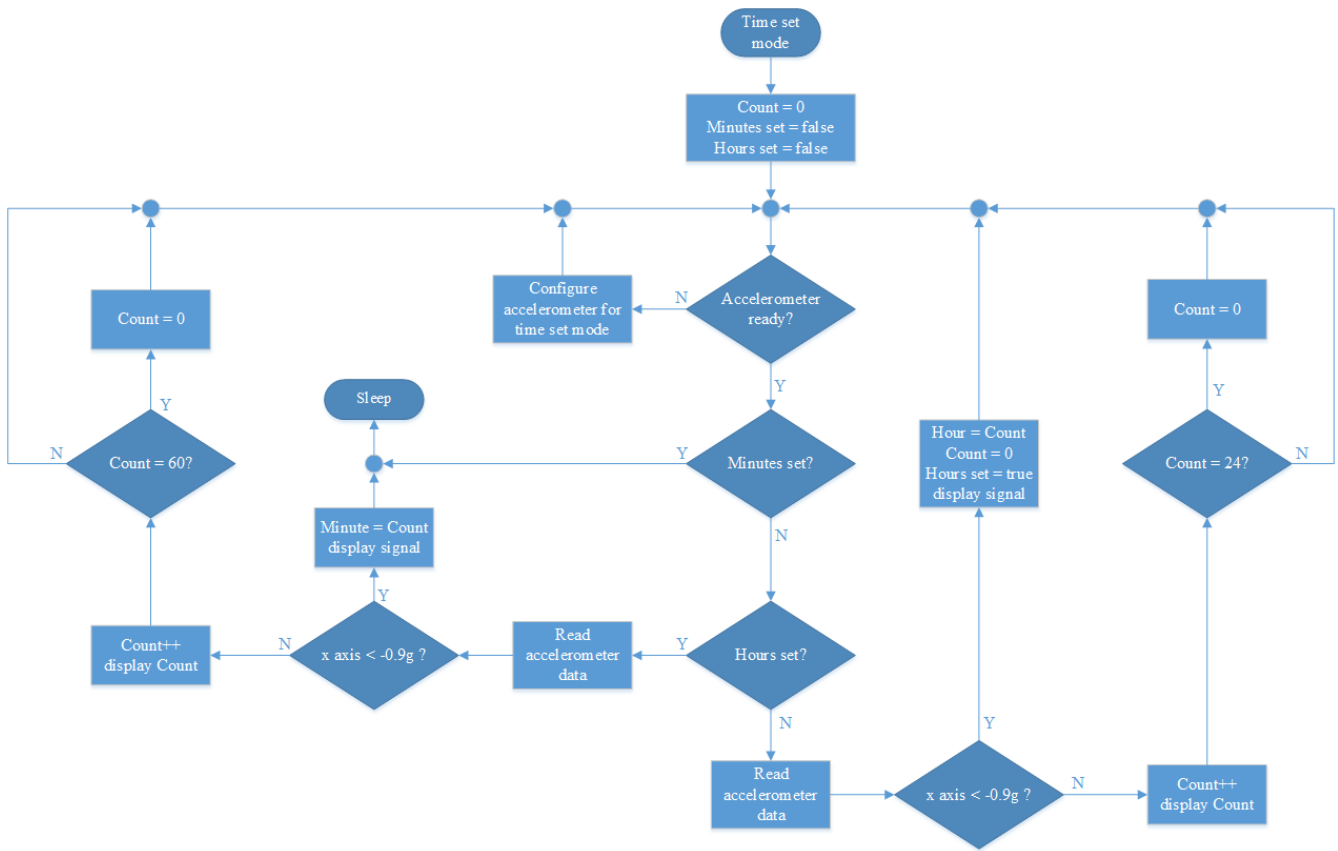


Figure 12: Flow chart for the time set module



Figure13: Gyro'clock displaying time. Variations in brightness caused by tumbling are most noticeable in the top right image

Results and significance

The Gyro'clock displays time at the top of the spin circle as expected when it is spun counter-clockwise in a circular path perpendicular to ground. Although the display is more easily seen in the dark, it is also visible on a well lit room with a plain background. Figure 13 shows the resulting time display and triggering accuracy of the Gyro'clock. Variations in brightness caused by the tumbling effect are noticeable. However, the tumbling effect does not significantly affect the ability to read the time.

The trigger point detection algorithm effectively eliminates multiple triggers within a single cycle. Observed trigger points are well within the acceptable 20° shift at moderate rotational frequencies (~ 2 Hz). Closer to the maximum rotational frequency for trigger detection (~ 4 Hz), shifts as much as 45° away from the top of the spin circle are observed. An accelerometer with a higher range will reduce this shift to acceptable levels by providing more data points for the trigger detection algorithm to work with.

When the device is spun clockwise, a person standing in front (looking towards the spinner) will see the time, while the spinner will see a mirrored image (see Figure 14).

The rotational frequency, where the Gyro'clock operates, is limited by the range of the accelerometer. The maximum range of the MMA8452Q is $\pm 8g$. When the rotational frequency is increased, the centrifugal acceleration experienced by the Gyro'clock also increases. At a certain frequency (the speed limit) the centrifugal acceleration will shift to a point where it does not satisfy the threshold condition of the trigger detection algorithm. At speeds on or beyond the speed limit the Gyro'clock is not able to trigger the display. The speed limit can be increased by increasing the length of the key chain since centrifugal acceleration is inversely proportional to the radius.

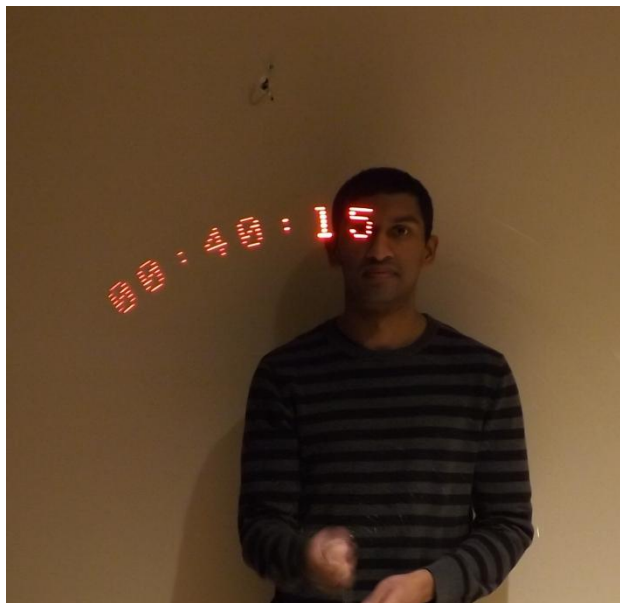


Figure 14: When the device is spun clockwise a person standing in front will see the time.

The significance of this project is in the challenges it presents. Through this project I have investigated a method to determine in real time when a spinning object is at a particular position along its circular path. Even though in theory it presented a simple problem, in practice it was necessary to investigate the nature of the problem further to achieve acceptable results. By observing and analyzing the forces acting on the device in real time, I was able to formulate a suitable method for detecting a special position along its circular path.

Through the formulation of an acceptable trigger point detection method, I have achieved the goal of this project by making a Gyro'clock, a portable POV display device that displays time when spun in the air by hand.

Further improvements to this project could be to use an accelerometer with a higher range, an LED arrangement with a 360° field of view to eliminate the tumbling effect, using RGB LEDs to add multiple colors, and more accurate trigger point detection methods.

Conclusion

Gyro'clock is a time keeping device that displays time when spun in a circular path by hand. A fully functional prototype of the Gyro'clock is successfully assembled and tested. An accelerometer is used to measure the centrifugal acceleration of the device, which is used to implement a triggering mechanism to display the time at the top of its circular path. The display is created by a row of seven LEDs scanning the image one row of pixels at a time. The brightness of the displayed image varies due to the tumbling effect, which can be reduced by increasing the diffusion of the LEDs. Time is set by using gestures, which are recognized using the accelerometer readings. Gyro'clock is rechargeable and runs for up to 40 days on a single charge. The maximum rotational frequency at which the Gyro'clock functions is limited primarily by the range of the accelerometer used.

Bibliography

1. Robinson, Paul, Rathod Ghansyam, Thakkar, Punit, Trivedi, Vishwa, "Persistence of Vision Control Using Arduino", *I.J. Intelligent Systems and Applications*, January 2014
2. Angesleva, Jussi, and Stephen Hughes. "Spin." N.p., n.d. Web. 15 Mar. 2015.
<http://www.luberth.com/help/spinning_keychain_led_clock_files/spinning_keychain_led_clock.html>
3. Carrol, Bradley, Ostlie, Dale, *An Introduction to Modern Stellar Astrophysics*. San Francisco: Addison Wesley, 2007. Print
4. Barta, D., Tibbitts, T., Bula, R., Morrow, R., "Evaluation of Light Emitting Diode Characteristics for a Space-based Plant Irradiation Source", *Advances in Space Research*, 1992, Vol 12, pp. 141-149.
5. Schutte, Herman, "Bi-directional level shifter for I²C-bus and other systems", Application note AN97055, Philips Electronics N.V., August 1997
6. "From Arduino to a Microcontroller on a Breadboard", Arduino, N.p., n.d. Web 15 Mar. 2015
<<http://arduino.cc/en/Tutorial/ArduinoToBreadboard>>
7. Turner, Clay. "Leaky Integrator." N.p., 7 July 2008. Web. 15 Mar. 2015
<<http://www.claysturner.com/dsp/Leaky%20Integrator.pdf>>
8. "Real Time Clock (RTC) Using the Asynchronous Timer", Application note 1259H-AVR, Atmel Corporation, January 2014