AVR Embedded Hexclock.

Overview:

This device is a fully functioning alarm clock. Rather than standard 24hr time, time is represented in binary as fractions of a day. 16 bits are used to represent the time in one day, and the time is displayed as a 16bit hexadecimal number. Internally, the current time is stored as a single 16-bit unsigned integer. The most significant bit (MSB) represents ½ of a day, the next most significant ¼ of a day, and so on, halving in value thanks to the binary system and it's use of powers of two. With 16-bits, the least significant bit (LSB) represents 1/65,536 of one day, and is what can be referred to as a hex-second or binary second since it approximates the duration of an ordinary second (which represent 1/86,400 of one day, being that there are 86,400 seconds per each 24 hour day) but is just slightly longer, approximately 1.318s or 86,400/65,536 seconds.

Since the MSB represents ½ of one day, this has the effect that the clock displays the time of noon as 0x8000, 0x0000 represents midnight, and 0xFFFF represents one hex-second or bit until midnight, after which the 16-bit integer overflows and it's value becomes once again 0x0000, the beginning of a new day owing to the properties of modular arithmetic.

Implementation:

The device is implemented in both hardware and in software. For this project I used an Atmel Attiny2313a AVR micro-controller, which uses Harvard architecture and a RISC instruction set. Nearly all features of the device are implemented using custom software, written in the C programming language and cross-compiled for the AVR architecture such that it can be burned to the chip's program memory by utilizing an ISP programmer. An 8.192Mhz crystal oscillator was chosen for use as an external clock source for the tiny2313a, as the internal RC oscillator is not precise enough for keeping accurate time. The chip however, runs at only 4.096MHz due to setting the clock prescaler in software to divide the clock frequency by a factor of 2.

Display:

Time is displayed using a fairly standard 4-digit common anode 7-segment blue LED display. The display part of the device utilizes an 8-bit (Texas Instruments TPIC6C595) SIPO shift register, allowing the display to receive data serially from the micro-controller and output a parallel data signal for interfacing the actual 7-segment display.. The shift register reads the high/low state of the the serial data line, using the clock and latch signal to shift in bits one at a time during each clock cycle, and once the register has been filled the latch signal is pulsed causing the stored bits to be output in parallel, appearing on the IC output pins simultaneously when it is triggered by the edge transition of the latch signal being pulled from logic low, to logic high. These signals and their timings are generated in software using a technique called bit-banging by controlling the states and timing of a few GPIO pins on the micro-controller.

The display is also multiplexed in software. Only one of the four digits of the display can be lit at any one time, so I utilized a multiplexing technique which cycles through each digit sequentially, utilizing code to handle another set of “chip select” signals and their timing; by using a few discrete transistors these signals allow the ability to select digits in sequence. The entire display is refreshed at a rate of 250Hz, which is much higher than the human eye can perceive so it appears as if all

digits are lit simultaneously due to persistence of vision.

When the device is reset, or first powered on, the display will blink and show all 0s, indicating that the clock has not yet been set. After setting the clock time, the display will no longer blink and the clock will begin keeping time.

Alarm:

This clock uses two features designed to alert you of the alarm. The alarm utilizes a small 2kHz piezoelectric buzzer to produce an audible chirping sound. The display also features a strobe effect to act as a visual stimulus. The alarm can be enabled/disabled via a slide switch, when enabled the alarm indicator light will be illuminated, serving to confirm that the alarm is in fact in the enabled state.

User Interface:

All features of the alarm clock can be interfaced via 3 pushbutton switches for setting the clock time and alarm time, and for activating the snooze. A slide switch is utilized to enable/disable the alarm. A blue indicator light near the top right corner of the display will be illuminated blue to indicate when the alarm is enabled. The snooze interval is optionally user-defined and can be programmed in if you find that preferable to the default snooze interval of approximately 8 minutes. By setting a jumper, you can select whether to use the user defined snooze interval, or the default snooze interval which is hard-coded into the firmware. User defined snooze settings are stored in EEPROM, which is non-volatile meaning that snooze interval settings will not be lost after a reset, power loss, or other power anomaly.

- Set/Select Key:
  - In normal operation, when this key is pressed the device enters set select mode allowing the time to be set. The digits can be set one at a time, indicated by the decimal point which will be illuminated next to the digit currently being set. Additional presses of Set/Select will allow selecting digits in sequence, once all digits have been sequenced through operation will return to normal operation, the time reflecting what was just set.

  - Which time is being set is dependent upon the state of the alarm slide switch when entered. If the alarm switch is set to alarm on, set/select mode will set the wake time for the alarm, otherwise if the alarm switch is off, set/select will allow setting the clock time. If the position of the alarm switch is toggled while in set/select mode, set/select mode will be exited discarding any changes as to not cause ambiguity about which time should be altered.

- Plus Key:
  - When currently in Set/Select mode, Plus key can be used to increment the currently selected digit, allowing each digit to be set independently.

- Snooze Key:
  - When the snooze key is pressed, the alarm buzzer is silenced for a period of time, usually about 8 minutes, unless you have set your own user defined interval. After the

the interval has passed, the alarm will sound again. The snooze key can be used as many times as needed.

- Alarm switch:
  - The alarm switch is a simple slide switch that either enables or disables the alarm. When enabled the alarm status light will be illuminated.

- Programming a user defined snooze interval:
  - To program the snooze interval, press and hold both the Snooze Key, and the Plus Key simultaneously for about ½ second until you see the word “prog” on the display. You can now let go of the snooze and plus keys. While prog is displayed, pressing Set/Select will take you to Set/Select mode which functions in the same way as for setting the clock time, only this time you are setting the interval of time between pressing snooze and the alarm resuming to emit sound, rather than an absolute time of day. A larger snooze interval means you will get to go snooze longer than with a shorter interval. Snooze interval can be set to anything between one hex-second to several hours or even nearly an entire day. User programmed snooze interval settings are stored in EEPROM which does not require power to maintain its state.

  - Note: Whether the clock uses the snooze you programmed in, or the default hard coded snooze interval is dependent on the position of the jumper. The jumper must be set to enable user-defined snooze in order to utilize this feature, otherwise the default interval will be used irrespective of any user-defined snooze interval.