To aid you in the difficult transition to consciousness, and help increase and maintain mental health
# Hardware Description

## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview</td>
<td>2</td>
</tr>
<tr>
<td>Power Supply</td>
<td>2</td>
</tr>
<tr>
<td>MCU</td>
<td>3</td>
</tr>
<tr>
<td>User IO</td>
<td>3</td>
</tr>
<tr>
<td>User Input</td>
<td>3</td>
</tr>
<tr>
<td>Keypad</td>
<td>3</td>
</tr>
<tr>
<td>Alarm View Button</td>
<td>3</td>
</tr>
<tr>
<td>User Output</td>
<td>4</td>
</tr>
<tr>
<td>Game/Menu Display</td>
<td>4</td>
</tr>
<tr>
<td>Clock Display</td>
<td>5</td>
</tr>
<tr>
<td>Audio</td>
<td>6</td>
</tr>
<tr>
<td>Time Keeping</td>
<td>7</td>
</tr>
</tbody>
</table>

## Overview

This document describes the hardware used to implement the Sleep to Wake Transition Catalyst. The Sleep to Wake Transition Catalyst is an alarm clock that requires the user to successfully answer arithmetic questions to turn the alarm off.

## Power Supply

The power for this device will be provided by a 5 volt 1.6 amp wall wart. This provides power to the device via a barrel connector with a 2.5mm inner diameter and a 5.5mm outer diameter. The jack for this connector will be located on my device. All of the components of the device run on 5 volts and are connected between the hot and ground provided by this jack.
MCU
The Sleep to Wake Transition Catalyst is built around the MC9S12DP512 microcontroller, and uses the following features.

- **PORTA**
  - PA0-PA7 (Out: LCD data bus)
- **PORTB**
  - PB0-PB7 (In: Keypad)
- **PORTK**
  - PK0-PK2 (Out: Control lines for LCD)
  - PK3 (Out: LCD backlight)
  - PK4 (In: Alarm View Button)
  - PK5-PK7 (NC)
- **SCI0**
  - TXD0 (Out: Speakjet)
  - RXD0 (NC)
- **SPI0**
  - SS0 (Out: LED Driver)
  - SCK0 (Out: LED Driver)
  - MOSI0 (NC)
  - MISO0 (Out: LED Driver)
- **IIC**
  - SDA (In/Out: RTC)
  - SCL (Out: RTC)

User IO

User Input
The user input of this device consists of two parts, a keypad and the alarm view button. The debouncing for both of these parts will be implemented with a state machine in software.

Keypad
The keypad is the primary input for the user. This is a sixteen button keypad arranged in a four by four configuration. The keypad has 8 output pins that are connected to Port B on the microcontroller. Four of these pins correspond to the columns on the keypad; the other four correspond to the rows on the keypad.

The column pins are connected to 5 volts using 10.2kΩ pull up resistors. When a key is pressed the row and column corresponding to the key are connected. If the pin connected to the row is low when this connection occurs, current will flow across the pull up resistor, causing the corresponding columns pin to go low.

By “scanning” or periodical pulling each row pin low and monitoring the column pins it will be able to tell if a connection has occurred, indicating a key press.

Alarm View Button
In addition to the keypad there will be one additional user input button. This is the alarm view button. This button operates on a similar principal to the keypad. One side of this button is connected to ground, and the other side is connected to pin 4 of Port K on the microcontroller. This pin has an internal pull up resistor and will be high when the button is not pressed. When the button is pressed this creates a path from PTK4 to ground and this pin will be low.
**User Output**
The user output will consist of three things: The game/menu display, the clock display, and the audio.

**Game/Menu Display**
This display is implemented with a 2x16 character LCD with a backlight. This LCD requires a total of 16 pin connections.

![Diagram of LCD connections](image)

**Power**
$V_{DD}$ is connected to 5V and $V_{CC}$ is connected to ground. These pins provide power to the LCD.

**Contrast Ratio**
The contrast pin determines the contrast between the background and the characters on the LCD. This pin can be set anywhere between 0 and 5 volts. Higher voltages correspond to a darker screen. This pin is connected to the arm of a potentiometer that is connected between 5V and ground. This will allow the contrast pin to be adjusted anywhere between 0 and 5 volts to provide an appropriate contrast ratio for the LCD.

**Control Lines**
The LCD has three control lines: the enable line, the read/write line, and the reset line. These control lines will be attached to pins 0-2 of Port K on the microcontroller. This will allow the microcontroller to control these pins via software.

**Data Lines**
Data is transmitted to the LCD over an 8 pin data bus. This bus is connected to the eight pins of Port A on the microcontroller. By writing different values to Port A the microcontroller is able to send commands to the LCD including writing characters, moving the cursor, and changing the display mode of the cursor.

**Backlight**
The backlight for the LCD is controlled from pin 3 of Port K on the microcontroller. This pin is attached to the gate of a MOSFET. The MOSFET is attached between the cathode side of the backlight LEDs and ground.
When the gate of the transistor (control pin) is low, the transistor blocks current from flowing through the LEDs. When the gate is high, the transistor provides a path to ground. Thusly the microcontroller can turn on the backlight by bringing the control pin high, and turn off the backlight by bringing this pin low.

The backlight specifies a forward voltage between 3.9 and 4.5 volts across the LEDs. Therefore the anode of the backlight is connected to 5 volts through a 5.1Ω resistor. At 120mA this resistor provides a 0.6 volt drop, supplying the backlight with 4.4 volts.

Clock Display

**LED Driver**
The LED driver is a multiplexed common cathode LED driver that can control up to 8 7-segment LED digits. One 10.2kΩ external resistor connected between 5 volts and \( I_{\text{SET}} \) is used to set the drive current for all LEDs.

**SPI Input**
The LED Driver operates as a slave on the SPI bus, and the microcontroller operates as the master on the SPI bus. The chip-select, clock, and data-in line on the LED Driver are connected to the slave-select, clock, and master-out-slave-in line on the microcontroller respectively. The fourth SPI line (master-in-slave-out) is left floating because the LED Driver is only written to, never read from.

The chip select line is active low. When the microcontroller pulls this line low, the LED Driver is able to receive communication, when the microcontroller has finished sending data it brings this line back high.

The data is transmitted from the MOSI (master-out-slave-in) line on the microcontroller to the Din (data in line) on the LED driver. The LED Driver reads this data on the rising edge of the clock signal. The clock signal is transmitted from the SCK0 pin on the microcontroller to the CLK pin on the LED driver at a data rate of 2MHz.

![Diagram of LED driver and microcontroller interface](image)

The microcontroller configures the mode, scan limit, and display intensity of the LED Driver via SPI. Once the LED driver has been set to the desired mode of operation, data is transited to the LED driver by first sending the number corresponding to the digit that you would like to write to (e.g. 0 for the first digit, 1 for the second digit), and then sending the decimal number that you would like to be displayed (e.g. 6 if you would like to display 6).
**LED Driver output**
The LED Driver accepts serial input over the SPI bus. The decimal value for each 7-Segment digit is stored in an 8x8 dual port static RAM. The data for the selected digit is then read from RAM and decoded by a BCD decoder. The decoded output drives the segment pins that correspond to the decimal number.

The LED driver is multiplexed so that only one digit is being driven at any time. This reduces the amount of power required to drive multiple digits, and also reduces the number of required connectors. The LED Driver “scans” the selected number of digits by successively pulling each digit pin low. Before pulling the selected digit pin low the segment pins are set are loaded from RAM though a BCD decoder as described above.

**7 Segment LED Display**
The clock display is a common-cathode four-digit seven-segment LED display. This is designed to work with a multiplexed driver. Therefore, it has just one input for each segment that is connected to that segment on each digit. The display provides one digit select pin for each digit (four in total). These pins are connected directly to the output pins on the LED Driver.

**Audio**
The audio output for the Sleep to Wake Transition Catalyst consists of five stages. The first stage is a natural speech and complex sound synthesizer called the Speakjet. The output of this is filtered with a low pass filter and passed through a potentiometer that controls the volume. The Output of this is then fed into an audio amplifier. The output of this is fed, finally, to the speaker.

![LED Display Diagram]

**Speakjet**
The Speakjet is preprogrammed with seventy-two allophones as well as pitch, speed, bend and volume control. Each allophone corresponds to an individual speech element. By stringing these allophones together and controlling the speed and bend of each it is possible to synthesize human speech. The Speakjet is also equipped with 43 sound effects.

The mode of the Speakjet is set with the mode lines, M0 and M1. The Speakjet is hardwired to “normal mode” by tying M1 high though a 10.2kΩ pull-up resistor, and grounding M0. The reset line is also tied high though a 10.2kΩ pull-up resistor because the device will not need to be reset during operation.
The Speakjet can be configured to take commands either through a standard serial port, “Event Inputs”, or both. For this device, the Speakjet is controlled serially and does not use the event inputs. Since the Event Inputs are not used, all of the Event Input pins are grounded as specified by the Speakjet datasheet.

Data and control commands are sent to the Speakjet using standard SCI protocol running at a baud rate of 9600 (the Speakjet default rate). Because the Speakjet is simply used as an output, only the transmit line from the microcontroller is used.

Commands sent to the Speakjet are stored in a sixty four byte input buffer. These commands are decoded and synthesised by the Speakjet. The output is a voice modulated signal on a 32kHz square wave carrier.

**Low-Pass Filter**
The output of the Speakjet is filtered by the low-pass filter recommended by the documentation.

**Volume Adjust Potentiometer**
The output of the low-pass filter is fed into a potentiometer. The output is taken from the arm of this potentiometer. The adjustment for this potentiometer is connected to a volume adjust knob that allows the user to adjust the volume.

**Audio Amplifier**
The output from the potentiometer is fed to the input of the audio amplifier. This audio amplifier operates using a single 5 volt supply. The amplifier is connected using the manufacturer’s suggested circuit.

**Speaker**
The output from the audio amp is fed to an 8 ohm speaker.

**Time Keeping**
The timekeeping functionality of the Sleep to Wake Transition Catalyst is carried out by an RTC (real time clock). For the timekeeping function, the RTC requires an external 32.768kHz crystal connected between X1 and X2.

**I²C Interface**
The RTC stores the time and date information in registers. Using standard I²C protocol, the microcontroller can set and read the time information from these registers. The I²C bus consists of two lines: the SCL(clock) line and the SDA(data) line. Both of these lines are open drain and are connected to the +5 volt source though 10.2kΩ pull-up resistors.

**Backup Battery**
The RTC also has a built-in power-sense circuit that detects power failures. When the RTC detects a power failure it automatically switches to the backup supply that is provided by an external +3V lithium battery. While in battery backup mode the timekeeping operation continues, and the device draws less than 500nA of current from the battery; because of this, the device will keep time for up to 10 years without main power and the user will not have to reset the clock if the device is unplugged or looses main power.