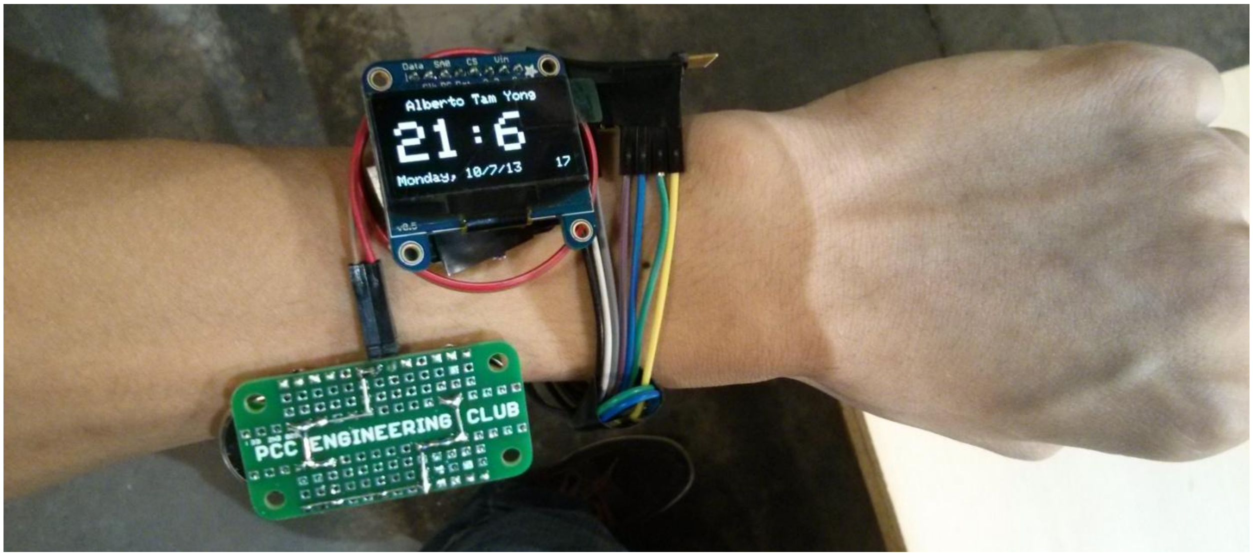
OLED Watch Project
Prototype 1
By Alberto Tam Yong
26 October 2013.



Abstract

The OLED Watch project consists of a compact Arduino-based system displaying time information from an OLED display. The project features the use of EDA software (EAGLE), PCB layout (EAGLE), SMD components, SMD soldering, and Arduino-based systems.

Introduction

Creating a digital watch presents many design challenges:

- The components have to be small enough to fit in a package that will rest on someone's wrist
- The whole system has to be light to rest comfortably on someone's wrist
- The components used have to be affordable since the system is meant for the simple task of showing the time

- The system has to be power efficient for the battery to last the most and be operational for longer than a day
- The system has to be reliable and solid to withstand day to day use

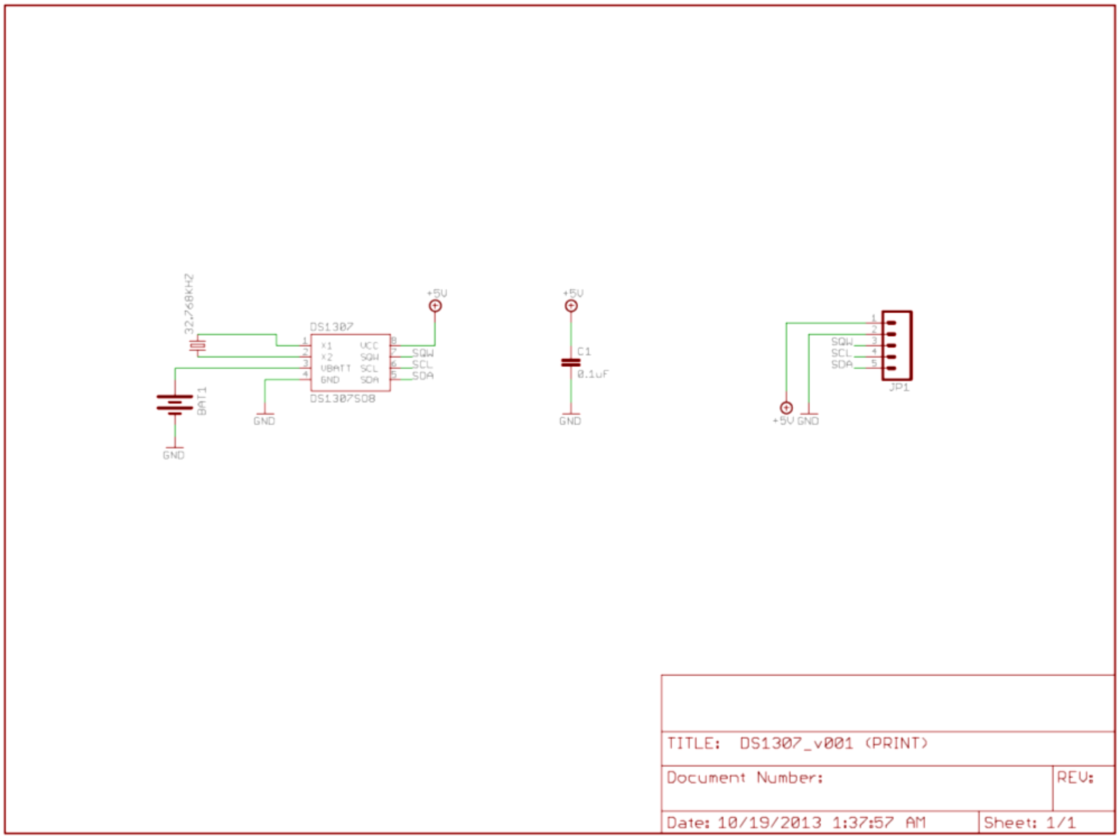
Materials and Methods

The Prototype 1 was made out of four main modules: an OLED display, a Real-Time Clock (RTC) module, an ATmega328p board, and a 6V power source. In addition, the ATmega328p was programmed using the Arduino IDE 1.0.1.

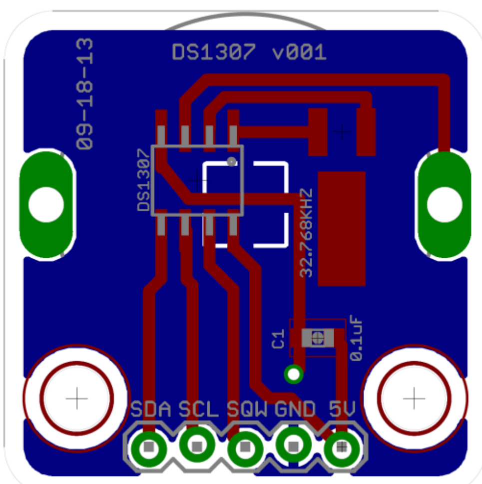
The Monochrome 1.3" 128x64 OLED Graphic Display was acquired from Adafruit.com. It was chosen for its ability to show basic graphics, power efficiency, flexible interface, and Arduino compatibility. The 128x64 resolution allows for the control of text and graph sizes. The numbers for the time are bigger in size to emphasize its importance while the date and seconds are in a smaller size. In addition, the placement and layout of the components on the display can be freely rearranged. The display has a nominal power consumption of 20 mAh at 5V, making it very power efficient due to the use of OLED technology. All the same, the display can be turned off when not in use for power saving purposes. Adafruit.com offers the display with two interface methods: I2C and/or SPI. The I2C interface was chosen for this project because of its minimal wiring requirement for its operation, two wires. Adafruit.com has also developed correspondent libraries to ease the operation of the displays: the Adafruit_GFX and the Adafruit_SSD1306 libraries. Using these two libraries, developing the code to control the display only requires a few lines of code and less time scripting the code.

Monochrome 1.3" 128x64 OLED Graphic Display

A self-powered Real-Time Clock module was designed for keeping track of time even when the system was not powered. The DS1307 real-time clock integrated circuit is at the core of the module. By adding a 2032 3V 255 mAh coin battery, the 1 in. sq. module has its own power source to keep track of time. The DS1307 uses the I2C protocol to interface with other devices. The Wire library available with the Arduino IDE allows for basic I2C communications. Some of the features of this integrated circuit include minimal circuitry, low power consumption, and the ability to keep track of time and date.



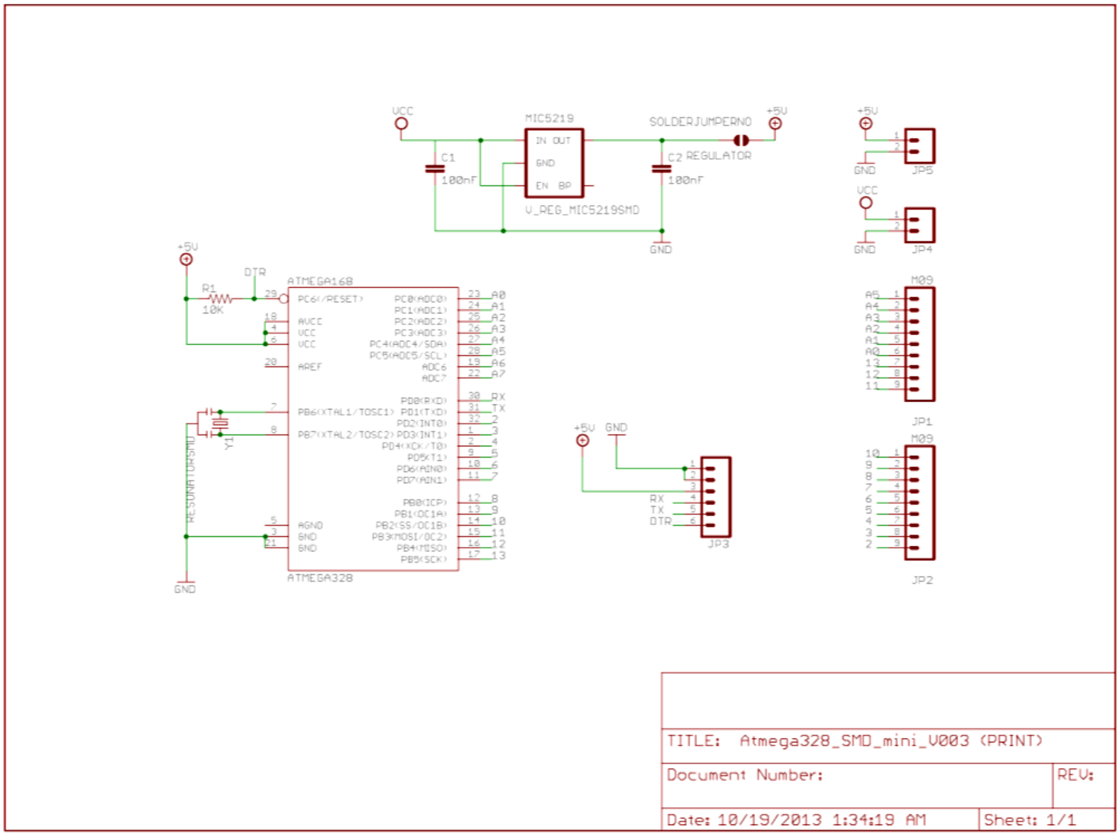
DS1307 Board - Schematic



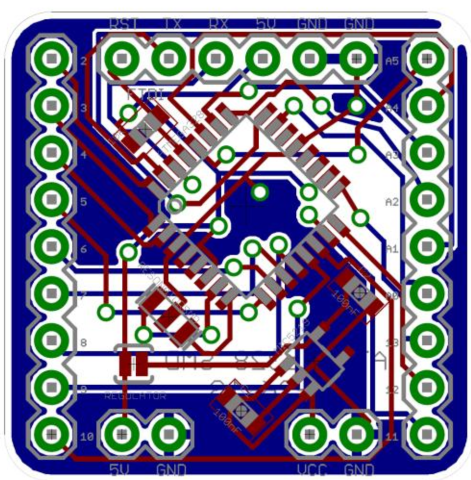
DS1307 Board - Layout

DS1307 Board - Assembled

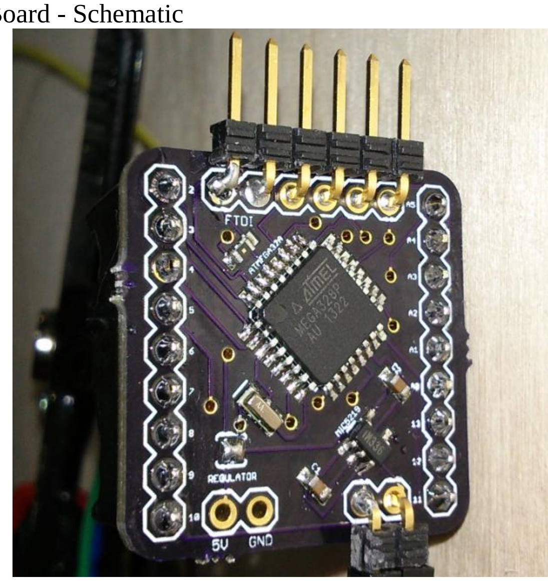
A custom ATmega328p board was used for the interpretation of signals and control of the display. The microcontroller ATmega328p was chosen for its low power consumption, wide variety of protocols, such as I2C, SPI, and UART, presence of interrupts, numerous GPIO pins, Arduino compatibility, 5V voltage operating system, and minimal required external circuitry. The custom board developed included a Low-dropout (LDO) regulator, SPX3819, and its correspondent circuitry to allow for a wide range of voltage input, a 16MHz resonator, and a designated header connection for programming the ATmega328p. All the component packages of the board are SMD components except the through hole header pins for the convenience of attaching headers, wires or any kind of connection. The 1 in. sq. board was designed to be compatible with solderless breadboards. Due to the use of SMD components, the board required the use of different SMD soldering techniques to be assembled. The ATmega328p integrated circuit, the main component of the board, was soldered by applying solder paste on the board's pads, placing the ATmega328p carefully aligning the pins with the pads, and using a heaterizer to melt the solder paste. Optimally, a hot air rework station, hot pan skilled or oven could have been used, but, due to the lack of proper equipment, the heaterizer was the best alternative. After soldering the ATmega328p with the heaterizer, a soldering iron with a very fine tip was used to retouch the soldered joins. The rest of the components were also soldered with the same technique even though it wasn't required to use this method. All the same, this method is time saving and efficient.



ATmega328p Board - Schematic

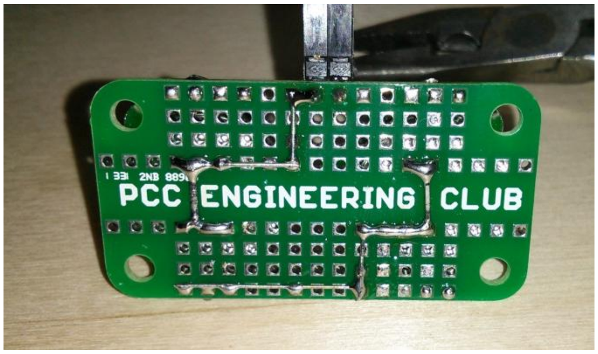


ATmega328p Board - Layout

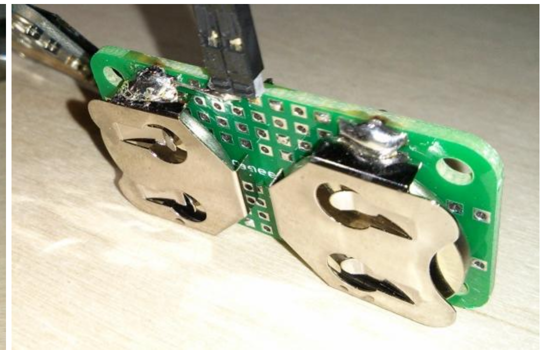


ATmega328p Board - Assembled

For power, a simple two 2032 3V 225 mAh lithium coin system was assembled using a custom perforated prototyping board. By soldering two coin battery holders and connecting them on series, the board was able to provide a total of 6V to the Low-dropout regulator (~300 mV dropout voltage) on the ATmega328p board. The coin battery holders were originally through hole components, but the leads were not standard, too big; hence, the leads had to be bend to turn it into an SMD component.



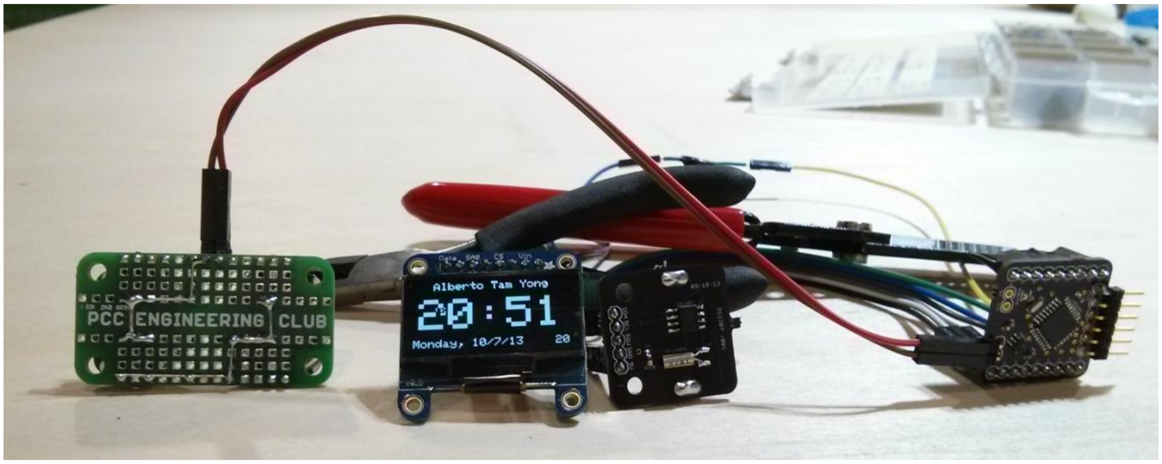
Custom Prototyping Board - Front


Custom Prototyping Board - Back

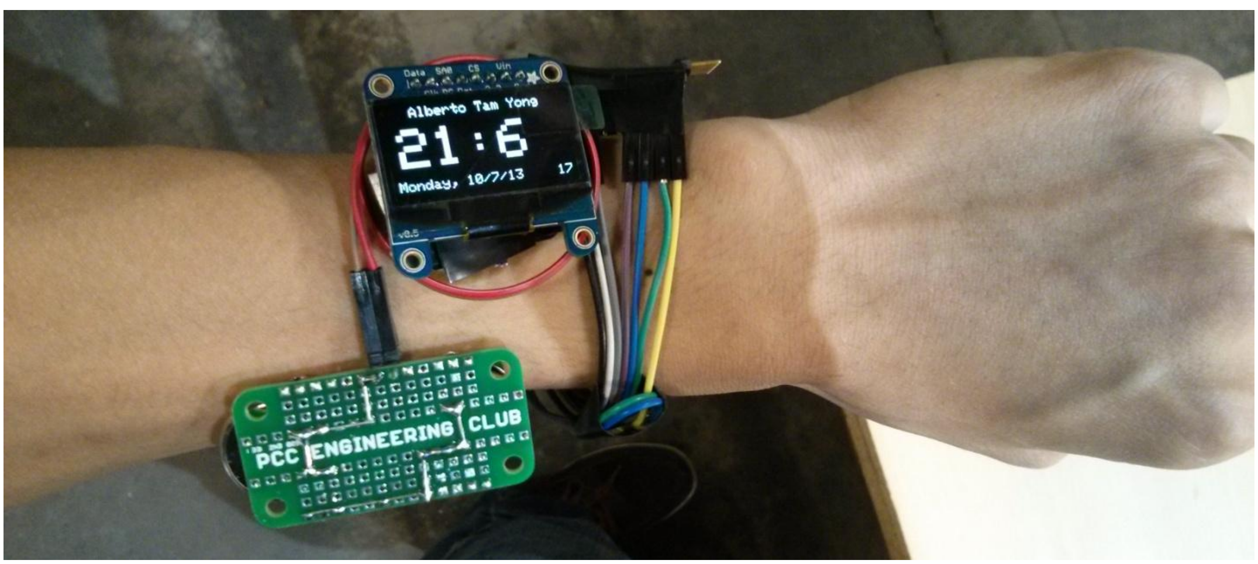
Bill Of Materials (BOM)

Qty	Item	Description	Price
1	ATmega328p		$3.16
1	Resonator 16MHz	SMD Package	$0.40
1	Resistor 10K	SMD Package 0603	$0.10
1	SPX3819 Voltage Regulator LDO 5V 0.5A	SMD Package SOT23-5	$0.99
3	Ceramic Capacitor 100nF	SMD Package 0603. Sold on a pack of 50	$0.09
1	Custom 1 in. sq. PCB Board for ATmega328p	Manufactured by OSH Park	$1.00
1	DS1307 RTC Clock Module	SMD Package SO8	$3.05
1	Crystal 32.768KHz		$0.28
3	Lithium Coin Battery 2032 225 mAh		$0.84
3	Coin Battery Holder 20mm	Thru-hole Package	$1.65
1	Custom 1 in. sq. PCB Board for DS1307	Manufactured by OSH Park	$1.66
1	Custom Prototyping PCB Board	Manufactured by Seeed Studio	$1.00
1	Monochrome 1.3 128x64 OLED Graphic Display		$24.50
9	Premium Female/Female Jumper Wires 6"	Sold on a pack of 40	$1.56
		Total	**$40.28**

All components of the OLED Watch displayed:



Model Scheme with Prototype 1:



For programming, the following code was used:

Arduino Code

```
#include <Wire.h>
#include <Adafruit_GFX.h>
```

```
#include <Adafruit_SSD1306.h>

#define OLED_RESET 13
Adafruit_SSD1306 display(OLED_RESET);

#define DS1307_ADDRESS 0x68
byte zero = 0x00;

#if (SSD1306_LCDHEIGHT != 64)
#error("Height incorrect, please fix Adafruit_SSD1306.h!");
#endif

void setup()   {
  Wire.begin();

  //Power Up RTC Clock Module
  pinMode(A3,OUTPUT);
  pinMode(A2,OUTPUT);
  digitalWrite(A3,LOW);
  digitalWrite(A2,HIGH);

  //Power Up OLED Display
  pinMode(12,OUTPUT);
  pinMode(11,OUTPUT);
  digitalWrite(12,LOW);
  digitalWrite(11,HIGH);

  // by default, we'll generate the high voltage from the 3.3v line
internally! (neat!)
  display.begin(SSD1306_SWITCHCAPVCC, 0x3D);  // initialize with the I2C addr
0x3D (for the 128x64)

  display.clearDisplay();   // clears the screen and buffer
  display.setTextSize(1);
  display.setTextColor(WHITE);
}


void loop() {
  printDate();
  delay(100);
}

byte bcdToDec(byte val)  {
  // Convert binary coded decimal to normal decimal numbers
  return ( (val/16*10) + (val%16) );
}

String dayOfWeek(int dayWord) {
  switch(dayWord) {
    case 1:
      return "Monday";
      break;
    case 2:
      return "Tuesday";
      break;
    case 3:
```

```
      return "Wednesday";
      break;
    case 4:
      return "Thursday";
      break;
    case 5:
      return "Friday";
      break;
    case 6:
      return "Saturday";
      break;
    case 7:
      return "Sunday";
      break;
    default:
      break;
  }
}

void printDate(){
  // Reset the register pointer
  Wire.beginTransmission(DS1307_ADDRESS);
  Wire.write(zero);
  Wire.endTransmission();

  Wire.requestFrom(DS1307_ADDRESS, 7);

  int second = bcdToDec(Wire.read());
  int minute = bcdToDec(Wire.read());
  int hour = bcdToDec(Wire.read() & 0b111111); //24 hour time
  int weekDay = bcdToDec(Wire.read()); //0-6 -> sunday - Saturday
  int monthDay = bcdToDec(Wire.read());
  int month = bcdToDec(Wire.read());
  int year = bcdToDec(Wire.read());

  //Show Time, Date, and Information
  display.clearDisplay();
  display.setTextSize(1);
  display.setCursor(16,0);
  display.print("Alberto Tam Yong");
  display.setTextSize(4);
  display.setCursor(4,16);
  display.print(hour);
  display.print(":");
  display.print(minute);
  display.setTextSize(1);
  display.setCursor(116,54);
  display.print(second);
  display.setCursor(0,54);
  display.setTextSize(1);
  display.print(dayOfWeek(weekDay));
  display.print(", ");
  display.print(month);
  display.print("/");
  display.print(monthDay);
  display.print("/");
  display.print(year);
```

```
  display.display();
}
```

The custom boards were designed using EAGLE software. EAGLE was used to draw the circuit schematics for the different boards and the PCB layout.

The custom board design were sent to manufacturing companies for them to be made. Some of the companies used include OSHPark.com and Seeed Studio Fusion PCB services.

Results

The OLED Watch Prototype 1 is functional and operational but have many issues. Some of the problems and improvements that could be made include power efficiency, battery life, size, casing materials, connectors, a Micro-USB port, use of a rechargeable battery, a cheaper microcontroller, board layout size, board material thickness, and a smaller and cheaper OLED display.

The custom ATmega328p board's linear voltage regulator is not greatly efficient since it steps down voltage by dissipating the extra electricity into heat. A more efficient voltage regulator such as a switching regulator or a boost converter with high efficiency would waste less power on heat and increase the power efficiency of the battery.

The whole system consumes too much power when all the devices are operational considering that it is a simple watch. The OLED display could be turned off when it's not necessary to display the time. The ATmega328p could be turned off or in a low power state when no processing or display is required. The battery could disconnected from any drain source when not in use.

The system could be simplified in less modular pieces. A possible combination could be the display, a battery, and the ATmega328p with the DS1307 and power regulators in a single board. The custom board could be attached to the back of the OLED display with minimal spacing, but without overlooking placing an electrical isolator, and the battery attached on the back of the custom board. Slots for the attachment of a strap could be added to fully resemble a typical watch.

The device could have a custom case made out of 3D printed materials. Plastic is a great candidate because it's easy to prototype, light, cheap, and convenient.

The interface to connect the multiple modules could be interlocking slide headers mounted flat on the custom boards to minimize any spacing between the boards. An example would be using right angle headers on the OLED display, an SMD female header on the custom board, and a JST 2-pin socket to connect a battery with a short cable. In addition, the Prototype 1 requires a 6-pin header to program the ATmega328p. An interface using a Micro-USB connector and an embedded UART to USB integrated circuit, such as the FT232, could be used to allow for changes on the code of the system.

A rechargeable battery would allow for a reliable power source for the system as well as a more flexible power source. The main alternative would be using a 1-cell LiPo battery with a nominal 3.7V voltage. However, by using a power source with less than 5V, which is the operational voltage of the system, requires to step-up the voltage by either using a switching regulator or a boost regulator. In addition, while using a rechargeable battery, an embedded system or module for charging the battery directly from the device would be the most practical approach for this design. An example of a 1-cell LiPo battery would be using a TPS61200 for boosting the voltage and a MCP73831 for charging the LiPo battery. In addition, a Micro-USB female socket could be added to allow for charging a rechargeable battery and for possibly updating or changing the firmware and software of the system.

Even though the ATmega328p is a very efficient and flexible microcontroller, the system does not require all the GPIO pins and interface protocols that the ATmega328p possesses. A smaller microcontroller, such as the ATtiny85, could be used to reduce power consumption, reduce the size of the overall system, decrease the cost of the device, and avoid wasting the additional processing power from the ATmega328p that is not required for this application.

The separate modules used for the Prototype 1 were not designed to maximize space and size layouts. An improvement could involve laying out all the custom board into a single composite board. In addition, the PCB thickness of the original modules are 1.6 mm. thick. This thickness could be reduced to decrease the size of the overall system. Flexible PCB boards could be ideal because of their thickness.

The OLED display offered by Adafruit.com, even though inexpensive, still represents the biggest cost on the budget. A less expensive display would allow for a smaller final product cost and price.

An additional detail about the OLED display used on the Prototype 1 is the refresh rate on the screen. When the device is shaken or quickly oscillated up and down, then image on the display gets distorted because of the lower refresh rate. To avoid this issue, a higher refresh rate should be set up.