Custom Smartwatch from Open-Source Design

University of Wisconsin Eau Claire

MULTI-APPLICATION WIFI AND BLUETOOTH-ENABLED SMARTWATCH BUILT FROM TTGO T-DISPLAY DEVELOPMENT BOARD

Spencer Keith | Department of Computer Science | Credit to Gabriel McFarlane (Minergabe on GitHub) | Advisor: Dr. Rushit Dave

TOOLS AND MATERIALS

HARDWARE

- TTGO T-Display development board
- 220 mAh, 3.7 V lithium polymer battery
- 6x6x6 tactile switches
- SS12F15 sliding switch
- DS3231 RTC module
- 30-gauge silicon wire
- 2017 MacBook Pro
- 3D printer
- Soldering iron
- Rosin-core solder
- Rosin soldering flux
- Heat shrink
- De-soldering pump
- Wire cutters
- X-acto knife

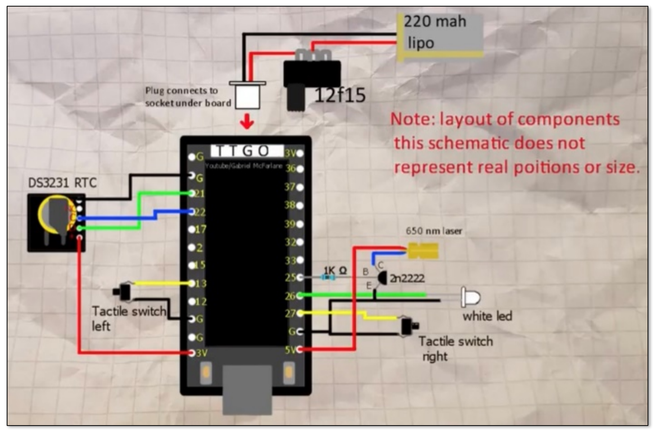
SOFTWARE

- C++ (language)
- Arduino (IDE)
- CLion (IDE)

HARDWARE CONSTRUCTION

- Removed attached connector from LiPo battery.
- De-soldered and removed plastic sockets from RTC module.
- Soldered board-included connector to battery and SS1215 sliding switch.
- Plugged connector into underside of development board.
- Soldered 30-gauge wire to underside of development board, RTC module, and tactile switches.
- Connected and soldered components together into finished circuit.
- Fit circuit into 3D-printed case.
- Sealed and glued cased shut.

Photo of smartwatch on wrist; prior to gluing, held together by zip-ties



Original schematic from Gabriel McFarlane, detailing the circuit layout

SOFTWARE DEVELOPMENT

- Started with original codebase from Gabriel McFarlane; deleted portions for hardware features *not* implemented in my build (flashlight, laser pointer).
- Reformatted code and reduced bloat from unimplemented features.
- Created Daily Checklist application. Added local SPIFFS text file to store data from each day for later use.
- Created Weather Tracker application. Registered an account with PirateWeather API to obtain API key. Implemented HTTP get requests to fetch data, and JSON parsing to parse data and display individual values.
- Created Network Settings application. Added first (and currently only) feature to scan for nearby networks and select one to connect to. Network passwords are stored on in a local SPIFFS text file cache and matched to the network ID during connection.
- Created Terminal application to display messages from the various watch functions that are performed during use. Rather than being navigated to, this application can be accessed at anytime by holding down both side buttons. Assists with debugging.
- Code is organized into most efficient possible functions and modules to assist with coding new functionality and debugging .

SOFTWARE DEVELOPMENT

MAIN DIGITAL CLOCK FACE

Displays the current time with military-format. Also displays device temperature and battery life.

STOPWATCH

Start and stop the stopwatch using the front right and left buttons, respectively. Accurate to the millisecond.

DAILY CHECKLIST

Numerically track the tasks you perform every day; in my case, feeding and taking my dog outside, and drinking cups of water.

WEATHER TRACKER

Fetches accurate and extensive regional weather data over WiFi using the PirateWeather API and HTTP get requests. Retrieves real and apparent temperature, precipitation, windspeed, and other data.

NETWORK SETTINGS

Scan for WiFi networks and select one to connect to. Uses a password cache for secure networks.

TERMINAL

When both side buttons are held down, the Terminal application opens, displaying a text interface with messages recorded during use (ex. HTTP errors)

We thank the Office of Research and Sponsored Programs for supporting this research, and Learning & Technology Services for printing this poster.