WATCHiT v3 technical document

9 november 2012, Emanuele Di Santo, lemrey at gmail.com

In this document we will briefly discuss the technical design and major factors that influenced the development of the third version of WATCHiT.

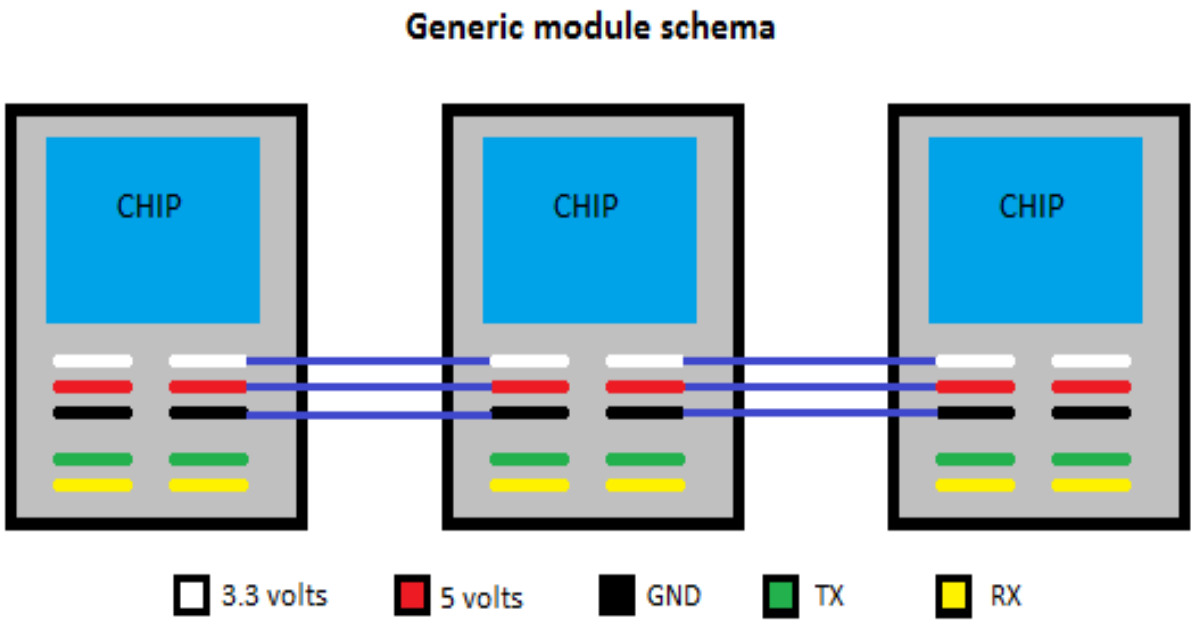
Overview

WATCHiT is a wrist-wearable, modular, Arduino-based device designed to be used in disaster management scenarios that
is capable of tracking user position and sending messages via wireless upon being issued commands using RFID tags. These messages include meaningful information as well as the user position at the time the command was issued.
It consists of various electronic components such as:

- a GPS device
- a vibrator motor
- a RFID tag reader
- an Arduino microntroller
- a Lithium polymer battery
- a Wireless communication device

Design

WATCHiT itself consists of five modules attached to a Velcro wristlet. The choice of a modular design was motivated by the necessity of having a device that could be actually
worn on a wrist. If the device consisted only of one necessarily bigger board it couldn't have been worn comfortably. The modules were assembled using perfboards which are inexpensive,
easily available and allow fast prototyping of simple boards. Effort was made to make them small and similar in order to keep the cabling simple and unobtrusive. We attached some Velcro
to the bottom side of each so that they could be fastened and unfastened many times during development, making the device easily disassemblable. A 3.7 volts rechargeable lithium polymer
battery, in short LiPo, was used to power the device. LiPo batteries are efficient, rugged, reliable, thin, have a very low self-discharge rate and good capacity. A 800mA battery
is able to power the device during normal operation for about 4 hours. Every module runs on 3.3 volts except for the tag reader which runs on 5 volts. A DC-DC step up integrated
component was used to supply the required voltage to the tag reader. Each module has two rows of five pins each, three act as a power bus and the remaining two are used for
serial communication. Having pins on both sides allows to rearrange modules on the wrislet in any order without cables messing up.



Modules

- **Main module**: Hosts the Arduino microcontroller, a transistor and three status leds: red, yellow and green. The Arduino board we used is an Arduino Pro Mini running at 3.3 volts.
  We picked this particular board because of its tiny size. The red led is a battery indicator: it will glow when the battery is low on power. The yellow led is a memory indicator.
  Every message sent by the device is in fact saved in the microcontroller EEPROM which has a capability of 15 messages. When this limit isreached, upon sending
  a new message, the least recent message will be overwritten and the led turned on. The messages can be later retrieved using a dedicated RFID tag.
  The green led is a GPS status indicator: it will blink when the device is receiving GPS data. When a GPS fix (the user position) is acquired the led will stay on.
  The transitor is a generic PNP transistor used to drive the vibrator motor. **Note**: in both prototypes, the leds lack any current limiting resistor. This is generally a bad idea.
  Each led should have its own current limiting resistor. Schematic

- **GPS module**: Hosts a **Fastrax UP501** GPS device, a white led and a coin battery. The white led is a GPS signal indicator. It will blink each time the device receives GPS updates,
  which is usually once per second, after a GPS connection is estabilished. The coin battery acts as a backup power supply for the GPS internal SRAM where the satellite data is stored.
  The backup power **must** be supplied to the GPS device. The positive lead of the battery should face the chip.
  Schematic

- **Xbee module**: Hosts a **Xbee S1 PRO** device for wireless communication. The device is configured to send and receive serial data at 19200 baud rate. The range is about one
  kilometer in open air. This module has a red and green led also. The red led will blink when data is being sent, the green one when data is being received. Schematic
  Note: in the first prototype built, the wiring scheme was incorrect and these two leds don't work, plus they lack a current limiting resistor.
  In the second prototype for space reasons, only one resistor was used. Two (one for each led) would be better.

- **RFID reader module**: Hosts an **ID-12** RFID tag reader device. Not much else to say.. A buzzer/led could be easily added but wasn't for space reasons. Schematic

- **Power module**: Consists of a **NCP1402** Sparkfun board. Attached on the LiPo battery to which is connected, regulates the voltage and provides 5v output.
  The first prototype had a super simple design with no on/off switch, which later appeared on the second prototype together with a red led. Schematic

- **Vibrator motor**: It's not properly a 'module' rather a small device that is placed under the wrislet at direct contact with the skin to provide feedback to the user.
  Because WATCHiT is supposed to be worn under a coat or suit, haptic feedback seemed the most natural choice. A buzzer could be used too, but in some
  situations the sound can't be heard clearly or it can be dampened by clothing so we opted for haptic feedback.
  The vibrator motor includes the needed 'safety' circuitry in order to avoid damage to itself and other components.

Implementation details

Wireless comm. is serial, at 19200 baud. Both sending and receiving devices must be set on that baud rate.
We distinguish between messages sent by the user and messages sent by the device autonomously (pings).
In example: **$P,User2,10:10:10,2000-10-10,63.41725,10.40284,message**
Messages are 64 characters (fixed) long strings, with seven comma separated fields:

- (1) type of message: $P for user messages and #P for 'ping' messages
- (2) username
- (3) time when the message was sent (hh/mm/ss)
- (4) date (yyyy/mm/dd)
- (5) latitude (5 decimals)
- (6) longitude (5 decimals)

- (7) string message (max 16 bytes).

Messages shorter than 64 character are filled with trailing spaces. Pings are sent by the device every 30 seconds after acquiring GPS signal.
They contain date/time informations and GPS position. Communication between Arduino and other modules is also serial at 9600 baud.

Future work and considerations

Several improvements can be made on future devices. Here we will list a few:

- small capacitors between each module Vcc and GND pins are desiderable in order to protect agains current spikes
- while the 5v output is regulated, the 3.3v is not, except for a small filtering capacitor
- as already mentioned, most leds on the modules lack a current limiting resistor
- pcb can be used to further reduce the size of the device
- the coin battery on the GPS module lacks a proper lock
- Xbee's antenna can be easily broken
- no kind of reverse polarity protection!