SMART WATCH CONTROLLED BY MOBILE PHONE VIA BLUETOOTH

Brigitta BALINT, Marian ALEXANDRU

”Transylvania” University, Brasov, Romania (brigi_balint@yahoo.com, marian.alexandru@unitbv.ro)

DOI: 10.19062/1842-9238.2017.15.1.6

Abstract: *The design and realization of a system similar to a smart watch is described in this paper. From a mobile phone, via an application created in Studio Android, notifications are transmitted to a system similar to a smart watch. Arduino Pro Mini is the basic module of this system and it connects the OLED (Organic Light-Emitting Diode) display and the Bluetooth Module HC-06 controller. After receiving and processing the notifications package, an individual page is assigned to each notification. The main purpose of the smart watch is to allow the user to see the notifications (emails, SMS, calls and current hour) without having to take the smart phone from the pocket .*

Keywords: *Arduino Pro Mini, Bluetooth, Android, Notification transfer*

1. INTRODUCTION

With the advancement of technology in the field of smart devices (phones, tablets, etc.) the user requirements also increased (buyers). For certain simple features user wishes no longer depend on direct interaction with the device (phone, tablet, etc.) but to be able to run on other devices, in order to see the incoming notifications such as emails, SMS, calls and current hour without having to interact with the smart phone. So, new types of smart devices that enable interaction between the user and the phone/tablet, smart watches and bracelets have been developed.

Mimo Loga (1941) is presented as the first smart watch. In 1972 came another smart device called Pulsar, which has been manufactured by the Hamilton Watch Company. "Pulsar" became a brand name, and in 1978 was bought by Seiko. It was able to store 24 digits, then Seiko started to develop smart watches able to store 2000 characters. Currently many electronic companies are producing smart watches. [1]

A smart watch can execute different functions, but the most important functions are displaying time/date, receiving and sending notifications from the smart phone and the ability to alert the user when receiving a new notification.

The system described in this paper was built like a smart watch but with greater size, consisting of a Bluetooth module HC-06, a Arduino Pro Mini plate, a monochrome display, a mobile phone (Android), keys and wires, etc. The developed app sends through Bluetooth from the mobile phone messages, emails, current date and time (Analog/ Digital format), in order to be displayed on OLED.

A very important feature of Bluetooth is to allow devices produced by different companies to work together.

2. SYSTEM IMPLEMENTATION

2.1 Equipment used. The Arduino Pro Mini, which has been chosen in order to reduce the size of the smart watch, is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM – Pulse Width Modulation - outputs), 6 analog inputs, an on-board resonator, a reset button, and holes for mounting pin headers. There are two types of Arduino Pro Mini: one at 3.3V and 8 MHz, and the other at 5V and 16MHz. In this application it was used the 3.3V variant. ATMEGA328 has 32KB memory for code storage, 2KB of SRAM and EEPROM 1kB.[2]

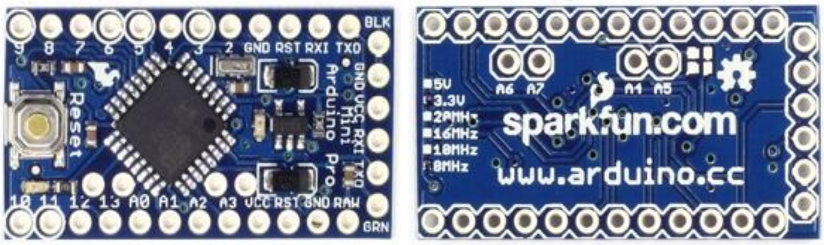

FIG. 1. The Arduino Pro Mini board used to develop the application

HC-06 module enables Bluetooth communication between devices and computers, mobile phones (Android, iOS).[3]

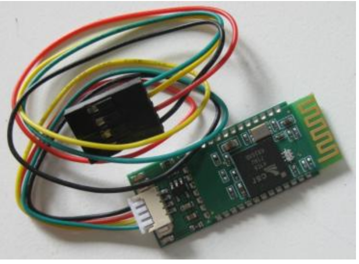

FIG. 2. HC-06 Bluetooth Module used for wireless communication between the smart phone and the Arduino board

The display used is small in size, has a diagonal of 1.3", but it is very easy to read, because of the OLED display. This display consists of 128x64 pixels, each of which is turned on or off by a controller chip. Because the screen produces its own light, no backlight is needed.[4]

2.2 Application description. The system block diagram is shown in Figure 3. The app on the mobile phone connects to the designed system via Bluetooth. Arduino Pro Mini is the core module, which binds all modules: OLED display, Bluetooth HC-06 Module, and the control key (button). The power supply consists of a battery. [5]

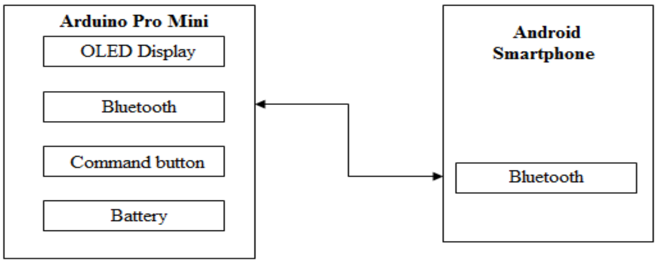

FIG. 3. System block diagram

OLED display uses I2C bus to communicate with Arduino. I2C supports multiple devices, and each device has a unique address on the bus.

The software is divided into two major parts. The first one is programming the Arduino module, which contains the microcontroller. Programming of this module was performed using the compiler provided by Arduino. The second part is the application software running on the mobile phone which was created in Android Studio. Images appearing on the OLED screen of the watch were created in Photoshop and then converted in array.

As mentioned above, the Arduino module receives data from the mobile phone using Bluetooth. The content of these data packets includes incoming messages, emails and calls from the mobile phone. In addition to these packages, Arduino receives the current hour from the phone.

Package = Call + SMS + Notification (Email)

The Arduino module receives each packet and should separate the calls, messages and emails. Once this separation has been accomplished, to each notification is assigned an individual display page, so it will result different viewing pages, as follows. Homepage (introductory page): displays the date, time, battery level and an alert if there was a new notification. Second page: displays received SMS notifications (sender and message content). The third page: displays notifications of incoming emails (sender and email content). The fourth page: displays the incoming call notifications (numbers/caller number or the total number of incoming calls if there are several unanswered calls).

Navigating from page to page is performed via the button on the clock. This button has dual functionality, detecting two events: short press or long press. Navigation is as follows:

If the initial page is the introductory then briefly pressing the screen causes the display to show notifications.

If briefly pressing the button is performed again, it scrolls between the pages of display notifications.

The return to the introductory page is accomplished by a long press of the button.

The distinction between a short and a long press is based on how much time the button is held down, the corresponding function being handle_button function ()

```
int handle_button()
{ int event;
   byte button_now_pressed = digitalRead(buttonPin); // pin High -> pressed
   if (!button_now_pressed && button_was_pressed) {
    if (button_pressed_counter < 50)
      event = EV_SHORTPRESS;
    else
      event = EV_LONGPRESS; }
   else
    event = EV_NONE;

  if (button_now_pressed)
    ++button_pressed_counter;
  else
    button_pressed_counter = 0;
    button_was_pressed = button_now_pressed;
  return event; }
```

At the end the function returns an integer set in compliance with the detected event:

EV_NONE = 0 if not detected touch of a button;
EV_SHORTPRESS = 1 if detected a short press of a button;
EV_LONGPRESS = 2 if detected a long press of a button.

The main task of the Android application is to retrieve notifications received by the mobile device, to process and transmit them via Bluetooth to the smart watch. [6]

In order to obtain notifications, the application uses a listening notifications service, registered at application startup.

```
    <service android:name=".NotificationListenerReceiver"
          android:label="NotificationListener"
android:permission="android.permission.BIND_NOTIFICATION_LISTENER_SERVIC
E"> [7]
```

The task of this service is to fetch notifications, processing and sending them to class CeasMain through intents.

The notifications are extracted in the status bar of the mobile phone. For each notification is extracted its title content and the package it belongs. Through the package it differentiates between various notifications types (email, SMS, missed call, etc.).

For example, if the package contains the string "android.gmail" it can be said that the notification received is due to an incoming e-mail. After the type of notification is determined, the information is transmitted to the smart watch.[7]

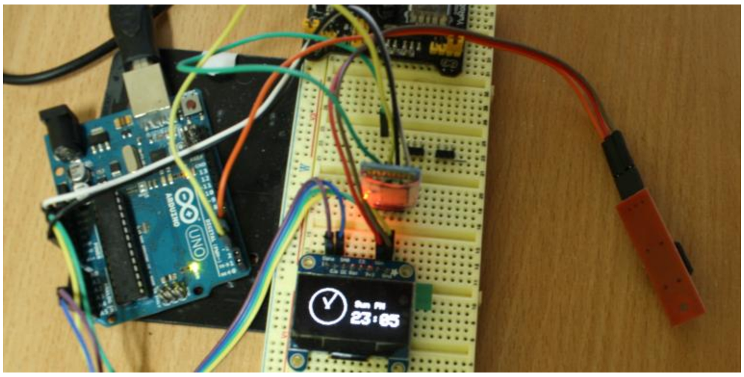

FIG. 4. The physical imlementation of the system

3. CONCLUSION AND FUTURE WORK

The system presented in this paper is the first step in designing a smart watch. One of the most important future development is to decrease the watch size and to extend the range of notifications that can be processed/displayed. To remove the physical button a color touchscreen LCD display will be chosen.

Regarding the Android app, it will be replaced with a widget that will have representation on the phone's home screen and will run in the background all the functions offered by the current application.

REFERENCES

[1] *** *Smartwatch*, Available at https://en.wikipedia.org/wiki/Smartwatch

[2] *** *What is Arduino Pro Mini*, Available at https://www.arduino.cc/en/Main/ArduinoBoardProMini

[3] *** *HC-06 Datasheet*, Available at http://www.fecegypt.com/uploads/dataSheet/1480849570_hc06.pdf

[4] *** *Monochrome 1.3" 128x64 OLED display*, Available at https://www.adafruit.com/product/938

[5] *** *Arduino Software (IDE)*, Available at http://www.arduino.cc

[6] *** *Bluetooth*, Available at https://www.bluetooth.com/what-is-bluetooth-technology/how-it-works

[7] *** *Android developer*, Available at https://developer.android.com/index.html