PROJECT DOCUMENTATION

Project Name : Android Watch

Team Members : Mayank Jain,Pranav Kumar,T Raghuveer,
Anurendra Kumar

Team Mentor : Sonu Agarwal,Swapnil Upadhyay

Basic Aim:

Creating a watch that connects to your Android Phone so that you Never Miss calls, messages or notifications using Arduino and LCD.

Motivation:

Think of the noisiest places, the quietest places and the busiest places you have been. How many times have you missed an important call, message or other notification because you couldn't hear the tone or it wasn't appropriate to be checking your phone?

A lot.

So, we with suggestions of our coordinators came up with the idea of making a watch that can connect to your smart phone and behave as a mini phone incase your phone is not accessible to you.

Theory:

Android watch reflects what is happening in your world and lets you know**.**

An Android watch is a computerized wrist watch with functionality enhanced beyond timekeeping, often with features comparable to a PDA. Many Android watches run mobile apps, some run on a mobile operating system, and a few have full mobile phone capability, while others function as mobile media players, with playback of FM radio and audio and video files, with sound via Bluetooth headset.

The Android watch we intend to make uses arduino and LCD shield connected to smartphone with the help of a bluetooth module and an android app.

Arduino is a single-board microcontroller designed to make the process of using electronics in multidisciplinary projects more accessible. The hardware consists of a simple open source hardware board designed around an 8-bit Atmel AVR microcontroller, though a new model has been designed around a 32-bit Atmel ARM. The software consists of a standard programming language compiler and a boot loader that executes on the microcontroller.

Bluetooth is a wireless technology standard for exchanging data over short distances (using short-wavelength radio transmissions in the ISM band from 2400–2480 MHz) from fixed and mobile devices and bluetooth shield is a device that enables UART communication of arduino to another bluetooth device.

The *LCD shield* is developed for Arduino compatible boards, to provide a user-friendly interface that allows users to go through the menu, make selections, display items etc.

Arduino:

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs.

The Arduino programming language is an implementation of Wiring, a similar physical computing platform, which is based on the Processing multimedia programming environment.

The Arduino development environment contains a text editor for writing code, a message area, a text console, a toolbar with buttons for common functions, and a series of menus. It connects to the Arduino hardware to upload programs and communicate with them. Software written using Arduino are called sketches. These sketches are written in the text editor. Sketches are saved with the file extension .ino. For our project we used Arduino Uno a microcontroller board based on the ATmega328.

Display Shield:

We used 4Display Shield 144 which provides an easy way of interfacing 4D Systems display modules to the Arduino-Duemilanove/Duo etc, the Arduino-Mega and many other Arduino compatible boards.

The 4Display-Shield-144 features a uLCD-144-G2 display module, a 5 way multiswitch joystick and male headers that help connect the 4Display-Shield to the Arduino boards. The 5 position joystick is connected to the Arduino D2, D3, D4, D5 and D6 pins. The communication interface between the 4Display-Shield and the Arduino is via the Serial UART. Both the TX and RX signals are jumpered on the shield PCB so that the Arduino UART can be freed-up when it's being programmed. The 4Display-Shield-144 is integrated with a full colour TFT LCD screen, tiny yet powerful

GOLDELOX graphics processors and a micro-SD connector which supports standard and high-capacity memory cards. The memory card can be used to store images, icons, video clips and data logging.

The display library used with arduino was displayshield4d after some changes in it.

The software used for working with the images on SD-card is Workshop 4-IDE which is quite easy to use.

Android:

Android is a Linux-based operating system designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Initially developed by Android, Inc., Android's user interface is based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching and reverse pinching to manipulate on-screen objects. The response to user input is designed to be immediate and provides a fluid touch interface, often using the vibration capabilities of the device to provide haptic feedback to the user. Internal hardware such as accelerometers, gyroscopes and proximity sensors are used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented, or allowing the user to steer a vehicle in a racing game by rotating the device, simulating control of a steering wheel.

In computer programming, Eclipse is a multi-language Integrated development environment (IDE) comprising a base workspace and an extensible plug-in system for customizing the environment. It is written mostly in Java. It can be used to develop applications in Java and, by means of various plug-ins, other programming languages including Ada, C, C++, COBOL, Fortran, Haskell, JavaScript, Perl, PHP, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It is the software we used for making our android app.

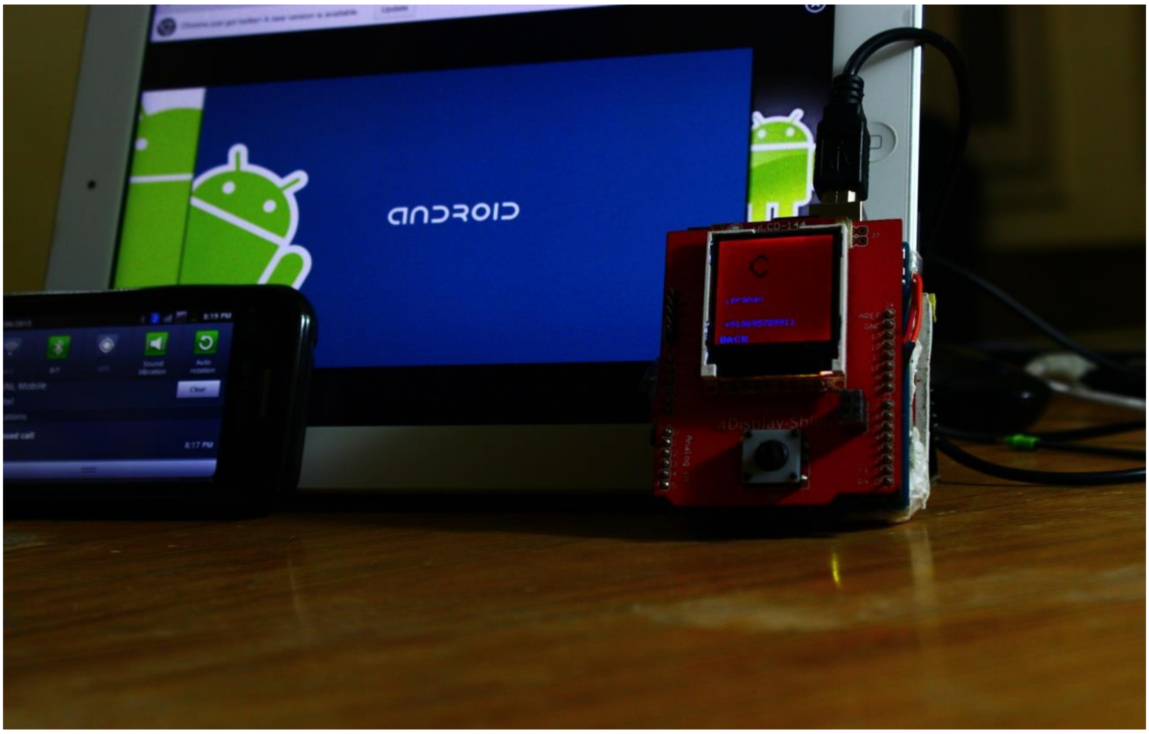
Overview:

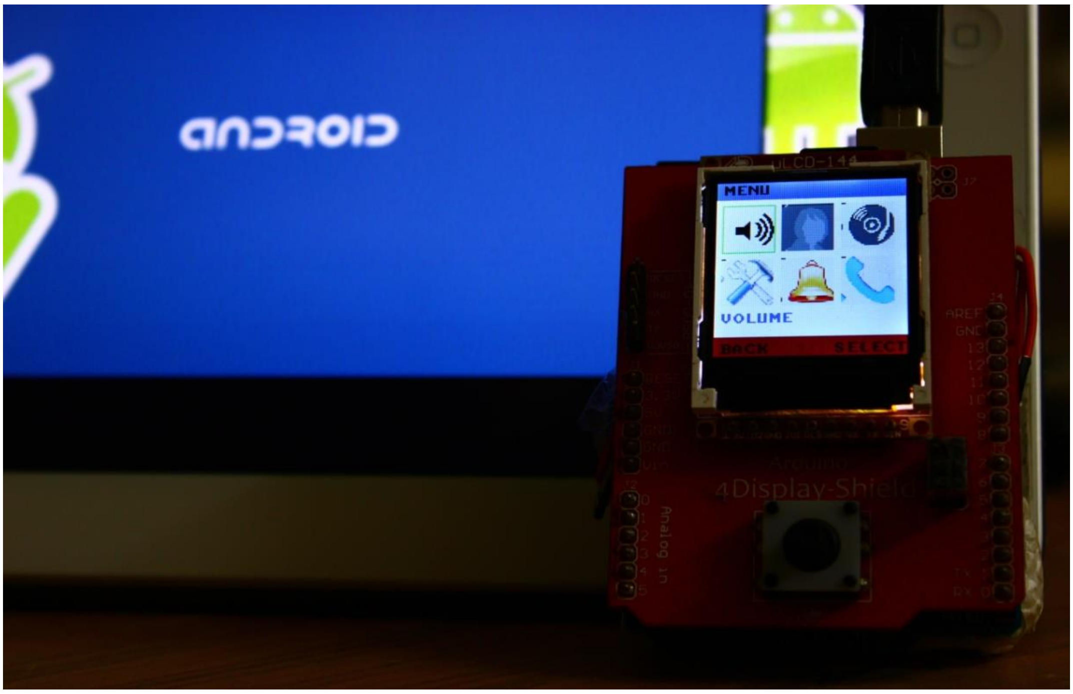
We started with Arduino and bluetooth module to establish a two-way serial-communication between android phone and arduino. As the Rx and Tx pins of arduino are to be used by display shield we used SoftwareSerial

library to create virtual serial ports at other pins. After that we started our next step i.e. studying android language and LCD module library.

After some time, we were able to Print the characters send by phone on the LCD display.

Then we started developing our android application that includes many features of android phone along with program for the Arduino. We wrote separate functions on arduino for various features of android phone supported by the app. Then we started our work to integrate those program and create a user-interface such as menu in the phone.

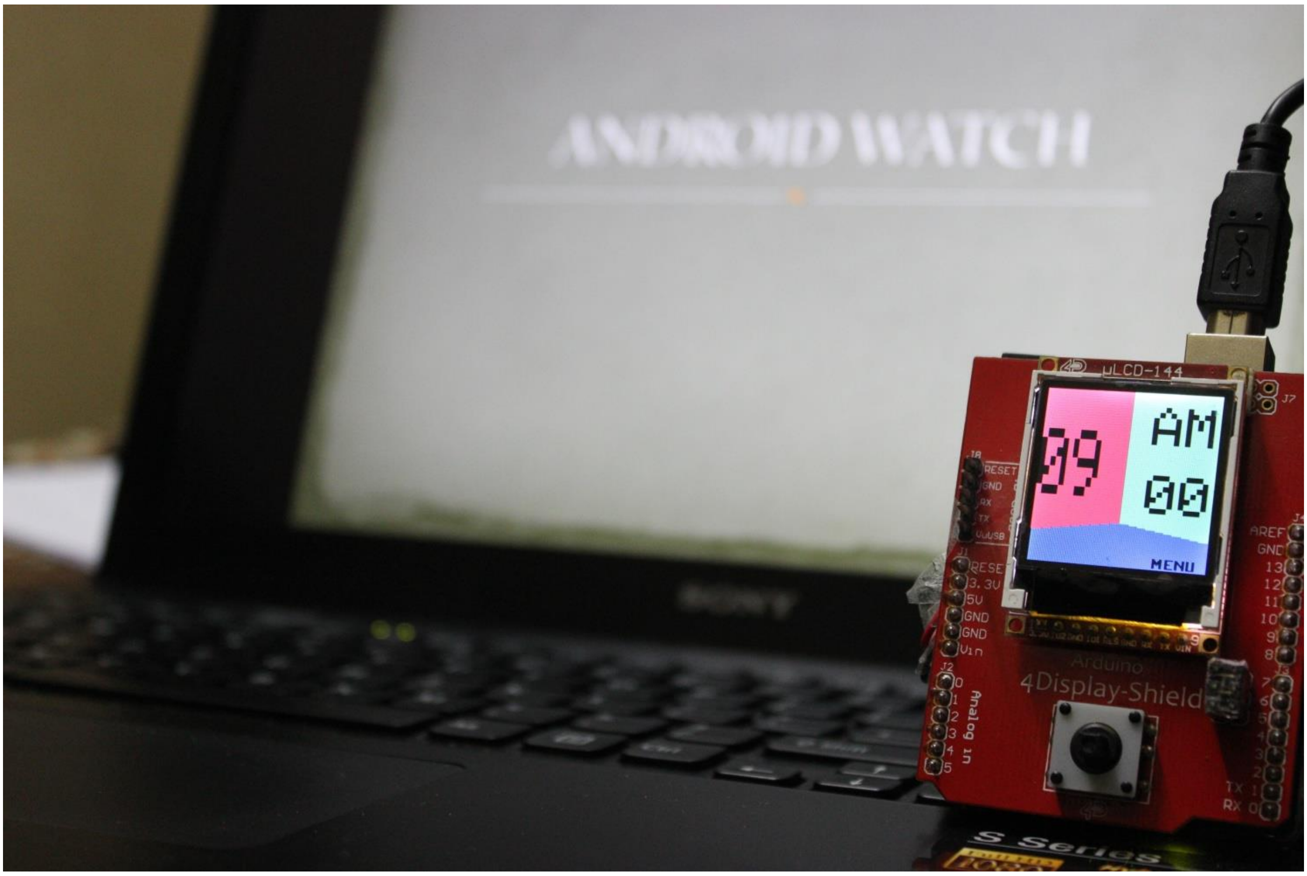




Implementation:

We started with Arduino Uno and RN-42 bluetooth module establishing a two-way bluetooth connection between the phone and arduino. We were able to create a connection between the two using sample app downloaded from google play (ex. Bluetooth SPP) but since we used u144 display shield which also uses the Rx, Tx (0,1) pins of arduino we had to use Softwareserial library to create virtual serial ports for bluetooth module to establish communication between arduino and phone.

After communication between the Arduino and phone, we started to work on display shield for which an arduino library (displayshield4d) is available that support features of drawing rectangle, circle and displaying alphabets.Pinchange Interrupts were used to create external interrupts on some pins as arduino uno has only two internal interrupts. But there is a limitation on using SoftwareSerial library with pin change interrupt that they can't be used simultaneously. The solution to this problem is quite simple just comment out the vectors which is not used in softwareserial library. We created individual programs displaying features such as messages, calls etc. we used several identifiers that identify whether it is a call or a message or any other feature. Suppose if there is a call, an identifier will be sent by the phone and arduino will recognize it as a call identifier and corresponding screen will appear displaying the call and an option to reject/accept the call. Then we started working on the interface of the display shield and integrating different programs. Since the display shield uses takes some time to process commands use of delay becomes very important. Images can also be displayed on display shield using Graphics Composer Tool in the Workshop4IDE software available for this shield.



For Android :

We used eclipse as the editor .We first read about Bluetooth . We can control basic operations of the inbuilt Bluetooth by the BluetoothAdapter class.We created an instance of this class and used it for connecting our android to the external Bluetooth module.

Connection between two Bluetooth devices can be established in two ways .1)Client side mechanism 2)Server side mechanism.

Connecting to other device by sending a request is called client side connection. Connecting to a device by accepting a request is called server side connection. Since Bluetooth module connects by server side mechanism we had to create an app that connects on client side mechanism. We referred to android developers.com and created such an app that connects by client side mechanism.

As we wanted our app to detect Calls, Messages we registered our app for the PHONE-STATE change and SMS received intent filter actions. We were thus able to retrieve Calls, Messages from our android phone and display them on our display shield.

We then took permissions for rejecting and accepting calls and implemented them. We simulated push up button actions for rejecting and accepting a call. We use querries to find the name from the retrived phone number. We read the code of desk clock and were able to set an alarm .WE send out an action intent for this purpose. We read about the Audio manager class in android. Instance of this class was created which could be used to control all the volume controls. We were partially successful in controlling the vibration of our android.

We then read about the music player app. We found the source code in GIThub. We were then able to control play, pause, next, prev functions of the base music player of the android. We could also retrieve callog from the phone and display it on our display shield. We then read about GPS, FLIGHT mode , WIFI. even in this case broad cast intents are used to toggle their states. We read about the location manager class. With this class we can find the location (latitude, longitude, altitude) and also the speed. We can further use the internet to find address accurate to street level, by using the location found. Flight mode can be switched on, but as we tend to loose our Bluetooth connection we can set an alarmtime at which it can automatically switch off the flight mode. Time and date was also retrived by the instance of calender class. We then worked on integrating all the features into a single app.

The Final Show :

- At the beginning, wallpaper is shown with the current time synchronized with the phone at instant of bluetooth connection.
- The next screen takes you to Menu screen which has many features including volume control, call log, alarm, music player etc..
- Volume control includes increase, decrease ,mute sound of function such as call, alarm, notifications.
- Call log displays last 9 call summary describing type of call by its colour.
- Alarm can be used to set alarm on your phone.
- Music player can be used to play/pause music, play previous/next song, increase /decrease volume of music.
- GPS can be used to switch on/off GPS .Further it can be used to determine location and speed of the android.
- Flight mode can be switched on and a time can be set such that flight mode is called off and automatic reconnection is established between watch and phone.
- Calls and messages will be displayed on incoming. Messages support scroll option.
- Profiles can be switched between general, vibrate and silent.

Further developments and Future Scope:

1). Use NFC to connect watch and phone.

2). Arduino uno and bluetooth module can be replaced by Arduino bt which has integrated bluetooth and quite smaller in size.

3). Replace bluetooth shield by a touch shield to make it a Android watch.

4). Adding more prominent features in the android application like heart rate, calories burn during exercises

Utility :

Your social network, your colleagues, family and friends know they can reach you because you miss nothing. Your phone can be in your bag, pocket or anywhere within 10 meters. Like a mini version of your smartphone, Android watch reflects what is happening in your world and lets you know.

Professional

At work, you can be discreetly notified of incoming emails, calendar events and other important information. Your smartphone can be in your pocket or bag.

Leisure

At the cinema, leave your phone in your pocket or bag. When something comes in, Android watch will let you know.

Urban

At a concert or other similar environment, it is easy to miss a call or notification. Android watch vibrates when something comes in. Trust it.