UMASS ECE Field Degree

# Baby Guard Midyear Design Review Report

Tony Tan, Kiran, Yun Shi

**Abstract—We introduce BabyGuard (a Wearable & Rich Featured Baby Monitoring and Safety System), a real-time system that will allow parents to remotely monitor the safety of baby. It can provide an alarm, at the most critical moment of babyhild safety has become a growing concern in recent years, despite advancements in safety equipment. There are several common concerns around children, which are: Missingmissing/kidnapping, ed children, Children children burned when playing with fire, or Heatstroke heatstroke when left alone in a car in summer, . It also enables parent and baby cannot to see/ / hear each other when far apart causing to ease anxiety. There have been a few systems developed to address these, with their limitations, that none of them is wearable and rich featured. With this in mind, and thanks to IoT, BabyGuard solves the solves the conflict between wearable and rich features conflict with a three parts architecture: a small wearable device + home server + Smartphone. A small wearable device, worn by a baby, through Bluetooth will connect a pre-paired parent's Smartphone or home server. When the wearable loses Bluetooth connection to all its pre-paired devices, Smartphone and the wearable both alarm. The wearable sends baby's ambient temperature to paired device periodically. Smartphone alarms if the temperature exceeds threshold. On connection of the wearable, the server will take video/audio of baby, and stream together with baby's temperature data to smartphone through WIFI. Smartphone alarms if temperature exceeds threshold. The server can also replay received video/audio of parents to calm down baby.connects and sends temperature data to a mobile phone if connected. If not, it checks the connection with the main server. If no connection is found, then an alarm sounds. The wearable device connects to the main server and sends temperature data periodically. The home server/base station connects to a mobile app, if available, through Wifi and sends gathered data .**

## 1 INTRODUCTION

Child safety has become a growing concern in recent years, despite advancements in safety equipment. There are several common concerns around children:

1) Missing/kidnapped children: Each year, there are 200,000 children kidnapped in China [1], and 800,000 missing in the USA [2].
2) Children burned when playing with fire: Every day, over 300 children between the ages of 0 to 19 are treated in emergency rooms for burn-related injuries and two children die as a result of being burned. Younger children are more likely to sustain injuries from scald burns that are caused by hot liquids or steam, while older children are more likely to sustain injuries from flame burns that are caused by direct contact with fire [3].
3) Heatstroke when left alone in a car in summer: On average, every 10 days a child dies from heatstroke in a vehicle. In more than half of these deaths, the caregiver forgot the child was in the car. A car can heat up 19 degrees in just 10 minutes. Unfortunately, cracking a window doesn't help. Young children are particularly at risk, as their bodies heat up three to five times faster than an adult's [4].
4) Parent and baby cannot see/hear each other when far apart causing anxiety

There have been a few systems developed to address these, notably:

1) Product A- Wireless Baby Monitor (Vvcare-851) [5], monitoring Ambient Temperature / Bidirectional voice and video interaction, yet only works according to fixed camera location, without distance alarm and not wearable.
2) Product B- Mini Smart Finder (Digoo DG-KF30) [6], which is a wireless Anti-Lost Locator for Kids with distance alarm and wearable, yet has no temperature sensor or camera for interaction. Too small to be carried by baby and may lead to suffocate

TABLE 1 the comparation of BabyGuard with other 2 products

| | Product A | Product B | BabyGuard |
|---|---|---|---|
| Ambient Temp. | X | | X |
| Body Temp. | | | X |
| Camera | X | | X |
| Interaction | X | | X |
| Distance alarm | | X | X |
| Wearable | | X | X |
| Cost | US$74.33 | US$ 4.71 | US$77.5 |

To begin the design of BabyGuard, we reviewed current solutions and their limitations. None of them is wearable and rich featured. With this in mind, BabyGuard solves the conflict with a three part architecture: a small wearable device + server + Smartphone.

A wearable device connects through Bluetooth and sends temperature data to a mobile phone if connected. If not, it checks the connection with the main server through Bluetooth. If no connection is found, then alarms. On connection, the wearable device sends to the paired device temperature data periodically.

The wearable is small and comfortable to wear on a baby's wrist, also safe with no loose ending parts. It has fewer sensors, so need smaller battery and can stand long time. Yet it provides the most vital features, prevent baby from missing or burnt/heatstroke

The home server/base station connects to a mobile app, if available, through Wifi and forwards temperature data, and exchanges video/audio. The server is interactive and always

Tony Tan Li, MA (email: tanli@umass.edu)
Kiran, MA (email: bhusanurmath@umass.edu)
Yun Shi, MA (email: yunshi@umass.edu)

connected system fixing at home. It's bigger than wearable, so is powerful and can cover most features of other competitors, and can be extended to many new fancy ideas.

Mobile app has a friendly UI, and alarms if baby's wearable is disconnected or ambient temperature exceeds threshold.

# 2 Design

## 2.1 Design Overview

The wearable device connects and sends temperature data to a mobile phone if connected. If not, it then checks the connection with the main server. If no connection found, an alarm sounds.

A wearable device connects to the main server and sends temperature data periodically.

A home server/base station connects to a mobile app, if available, through Wifi and sends gathered data.

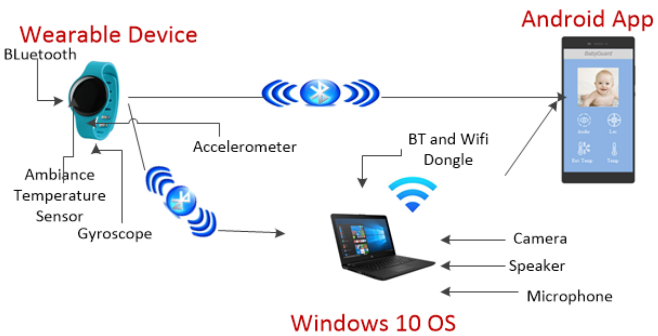


Fig. 1. Overall architecture

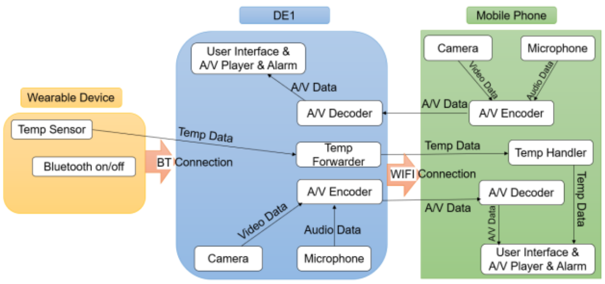


Fig. 2. Block Diagram

## 2.2 Design of the Home Server/DE1 board (PC is used to ease dev for MDR)

The home server is to be installed inside the baby's room and connects to a wearable device via Bluetooth and to the mobile phone via Wifi. It includes capabilities for Bluetooth and WIFI and camera, microphone, screen and speaker components. When the wearable device connects the home server through Bluetooth, the server activates its camera and microphone. The server sends video/audio of the baby to the parent's smartphone (for example in other room/office) through WIFI. The wearable device sends the baby's ambient temperature through Bluetooth to the server board, and the later forwards the information to the parent's smartphone through WIFI. The smartphone sends video/audio of the parents to the board, which replay on a screen and speaker to calm the baby down.

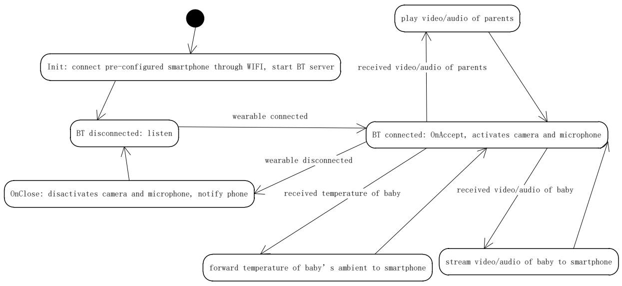


Fig. 3. State Chart (DE1)

We used a laptop instead of a DE1 board to ease development for MDR demo.

1) Before MetaBase wearable was purchased, MFC (Microsoft Foundation Class) prototype code was developed on a Windows PC:
   a) When Macbook connects to the PC with BT, void CSockSvr::OnAccept(int nErrorCode) is called.
   b) When Macbook power off its BT, void CSockClt::OnClose(int nErrorCode) is called.

More deliberate test on distance was done with result:

At distance 6~10 meters, after about 20 seconds, CSockClt::OnClose is called.

20 seconds is maybe the timeout of TCP, which is too long for alarm.

2) We designed to alarm when temperature data is missing for 6 seconds, which should be received at 2 seconds' interval.

To study socket programming on the same PC, we created server and client projects with normal loopback sockets. The client increases an integer as faked temperature data every 2 seconds and sends to server, from 8, 9, 10, 11, 12...

3) After MetaBase wearable was purchased, to verify it works with win10, we installed MetaBase win10 SDK (including MetaBase App and sample code), and got temperature data on win10 from Metawear as following:

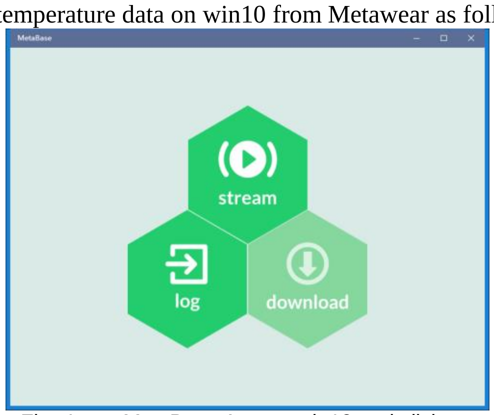


Fig. 4. run MetaBase App on win10 and click stream

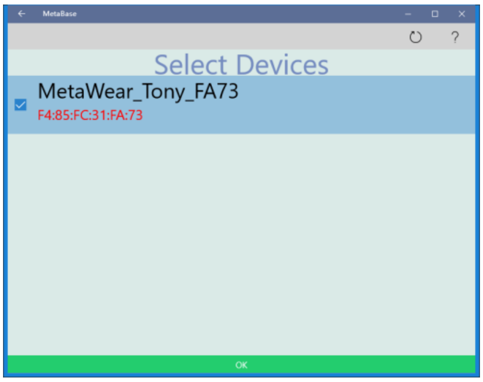


*Fig. 5.* select device

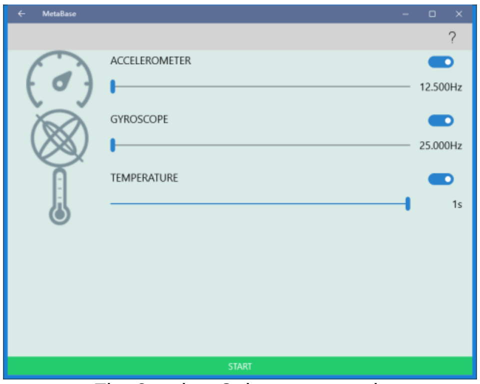


Fig. 6. select 3 data types and start

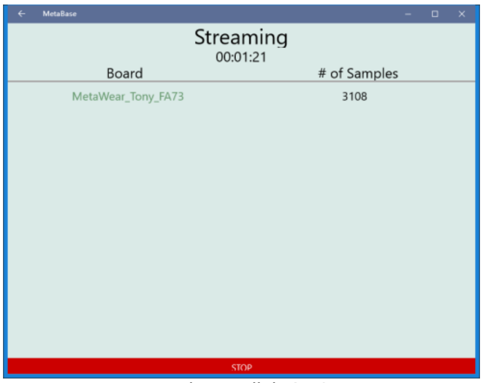


Fig. 7. click STOP

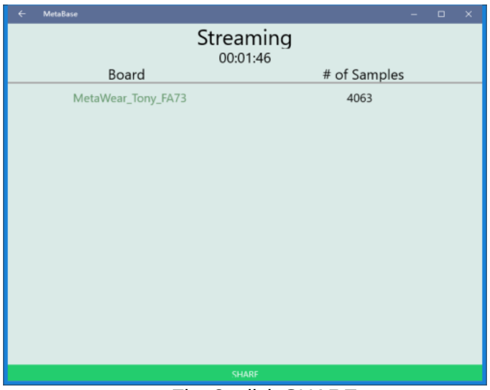


Fig. 8. click SHARE

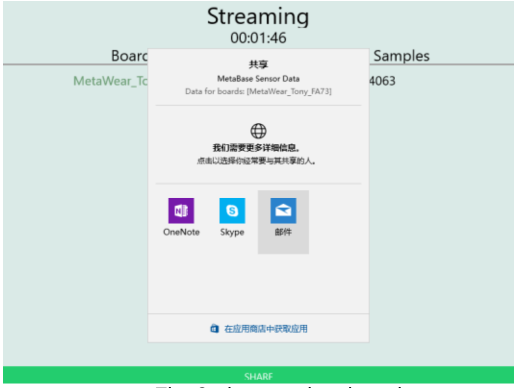


Fig. 9. data can be shared

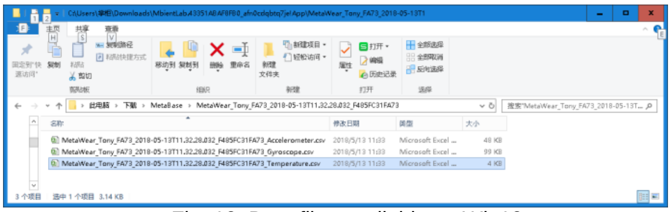

Fig. 10. Data files available on Win10

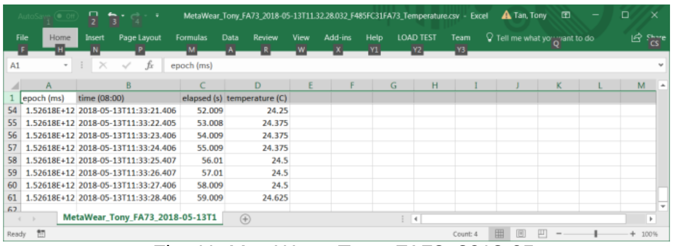

Fig. 11. MetaWear_Tony_FA73_2018-05-13T11.32.28.032_F485FC31FA73_Temperature.csv

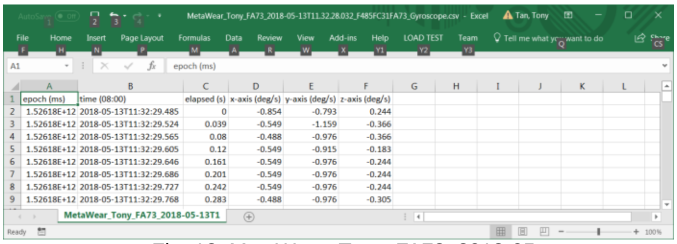

Fig. 12. MetaWear_Tony_FA73_2018-05-13T11.32.28.032_F485FC31FA73_Gyroscope.csv

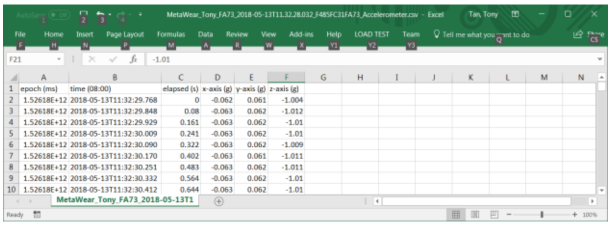

Fig. 13. MetaWear_Tony_FA73_2018-05-13T11.32.28.032_F485FC31FA73_Accelerometer.csv

## 2.3 Design of Wearable

The wearable is a small and compact device worn on the wrist or ankle. It connect via Bluetooth to the mobile phone or home server and transmits temperature data to these components. The wearable device sends panic signals if the baby's ambient temperature crosses a threshold. Alarm is sounded if the Bluetooth connection is broken because the mobile phone and home server not reachable by flashing light.

Detailed Metawear sensor product specifications [7], Metawear Documentation [8], Developer SDKs for the sensor and board integration [9], and Tutorial and training materials on code usage [10] are all available on mbientlab's website.

TABLE 2 the specification of wearable

| | |
|---|---|
| Dimension | – Diameter: 0.94in / 24mm |
| | – H: 0.24in / 6mm |
| Weight | – 5.6gms |
| Connectivity | – Bluetooth LE 4.0 – 2.4Ghz |
| | – Up to 100ft of range – typical 10m |
| | – Stream sensor Data from 1 Hz to 100 Hz |
| | – Log sensor Data from 1 Hz to 400 Hz |
| Core | – Nordic nRF51822 32-bit ARM® Cortex™ M0 CPU |
| | – 256kB/128kB flash + 32kB/16kB RAM |
| | – RGB LED |
| | – Micro push-button |
| | – I2C + 4 GPIOs |
| Logging / Memory | – Streaming ONLY Sensor |
| | – 80KB FLASH Memory – On board |
| | – 5K – 10K sensor data entries with timestamp |
| Battery | – 200 – 220mAH CR2032 Coin cell battery |
| Accelerometer | Accelerometer (A): 16 bit |
| | ±2g: 16384LSB/g |
| | ±4g: 8192LSB/g |
| | ±8g: 4096LSB/g |
| | ±16g: 2048LSB/g |
| | Noise density (typ.): 180 µg/√Hz |
| Gyroscope | Gyroscope (G): 16bit |
| | ±125°/s: 262.4 LSB/°/s |
| | ±250°/s: 131.2 LSB/°/s |
| | ±500°/s: 65.6 LSB/°/s |
| | ±1000°/s: 32.8 LSB/°/s |
| | ±2000°/s: 16.4 LSB/°/s |
| | Noise density (typ.): 0.008 °/s/√Hz |
| Temperature | Thermistor (Degrees C): 16bit |
| | ±0.5°C, Accuracy |
| | ±0.1°C Resolution |
| | -40…85°C Range |

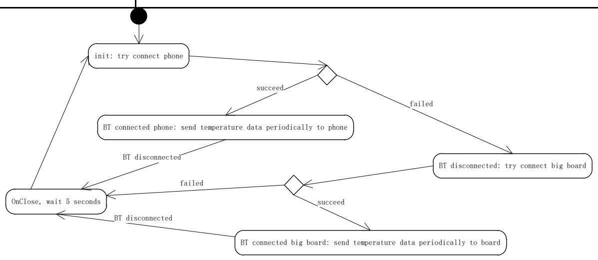


Fig. 14. State Chart (Wearable)

## 2.4 Design of APP

The Android application provides an easy to use, clean, and visually appealing interface for the user to interact with. Some of the main goals of the app design are to represent data in a friendly fashion, show progression over time, and provide feedback. The app includes views for temperature and graphs of history temperature data for a single wearable device. It can also indicate the motion condition received from the sensor. It is equipped with a menu screen to provide navigational functions. Web socket technology has been chosen to allow for communication between the mobile application and a computer application. This is a versatile, platform-agnostic protocol that is supported on many platforms. More importantly, it allows two-way communication in an arbitrary manner either party can send a message to the other at any time.

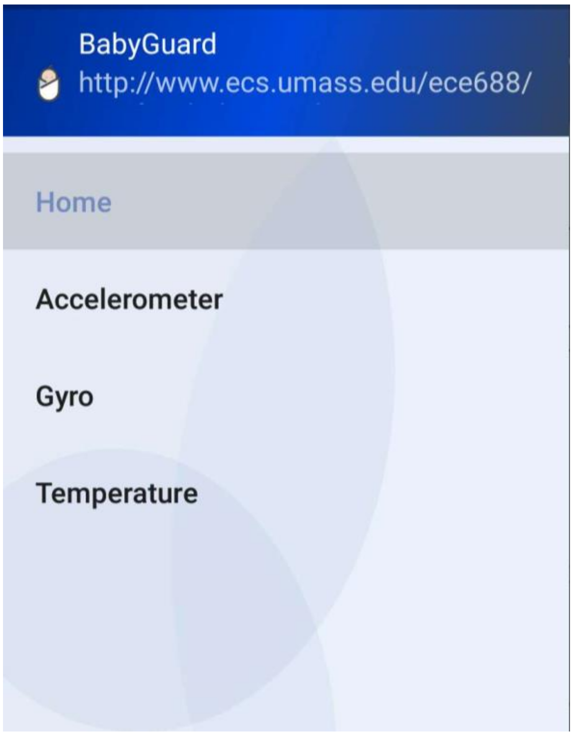


Fig. 15. User Interface of the APP

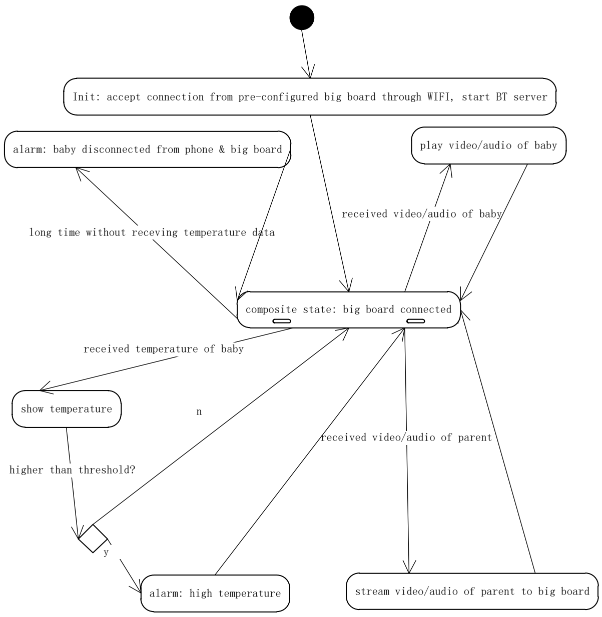


Fig. 16. State-chart of the smartphone

Above “composite state: big board connected” is detailed as following:

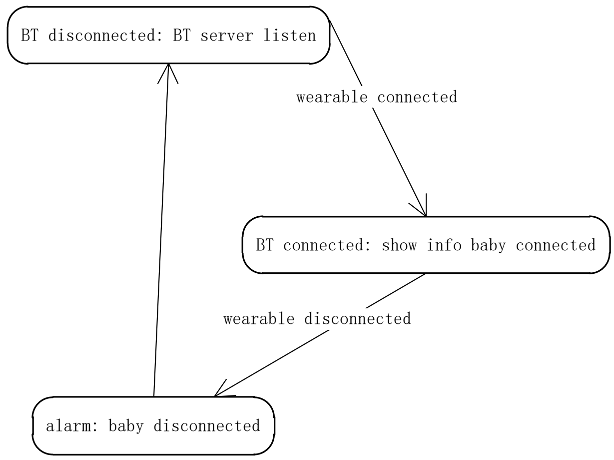


Fig. 17. composite state: big board connected of the smartphone

To create the application, an understanding of Android application development is required, which is something that must be learned. Specifically, development will be done in the Android language using Android Studio. This was chosen as it is modern and well-supported, targeting interactive application development of all kinds. Software development techniques learned in the past have been and will be useful in learning this new platform and problem-solving during the development.

The computer will run a server to establish a connection with the application over a network. This web socket server is created running on the node.js JavaScript environment. It listens on a port to establish a link with clients, in this case the application. The socket.io engine was used to implement this functionality. In addition, to support running the server with node.js, there is a socket.io library available for Android development which was integrated into the mobile application.

The Accelerometer and Gyro pages show the motion condition for the wearable device by tracking the movement amplitude of the wearable. Every time slot the movement created by the wearable device will create a data point and plot a point on the app 2-dimensional coordinate plot. The most recent dot will indicate whether the baby is moving and the trend will let the app user know whether the baby is asleep.

The Temperature page provides the historical temperature data that has been received from the sensor. It also plots a graph with which a user can easily track body temperature since normally there are minor fluctuation in a healthy person’s temperature. For a baby, temperature changes can indicate if medications have kicked-in.

The app also provides functions to connect and disconnect with wearable devices and supports device functions to change the LED display based on user preferences. To enable a user to quickly use the device, an embedded help function provides simple how-to guidance.

The future functionality of this app may include a video/audio interface with the home server, which would enable parents to chat with the baby and monitor the baby’s status. This function would be attractive since a face-to-face interaction will pacify the baby when parents are not around.

## 3 Project Management

The MDR goals for the BabyGuard project are listed below:

1) Establish connection between the wearable and laptop
2) The wearable sends temperature data to laptop
3) The laptop forwards temperature data to the smartphone
4) Smartphone shows temperature data in GUI

We have accomplished most of the above except for goal 3. The tasks that remain for the CDR are:

1) Implement alarms when connection is lost between wearable and phone
2) Wearable device connects main server and send temperature data periodically.
3) Home server connects to mobile app if available through Wifi and send gathered data.
4) Phone alarms if temperature above threshold
5) Server streams video/audio with smartphone
6) try to transition from a laptop to the DE1

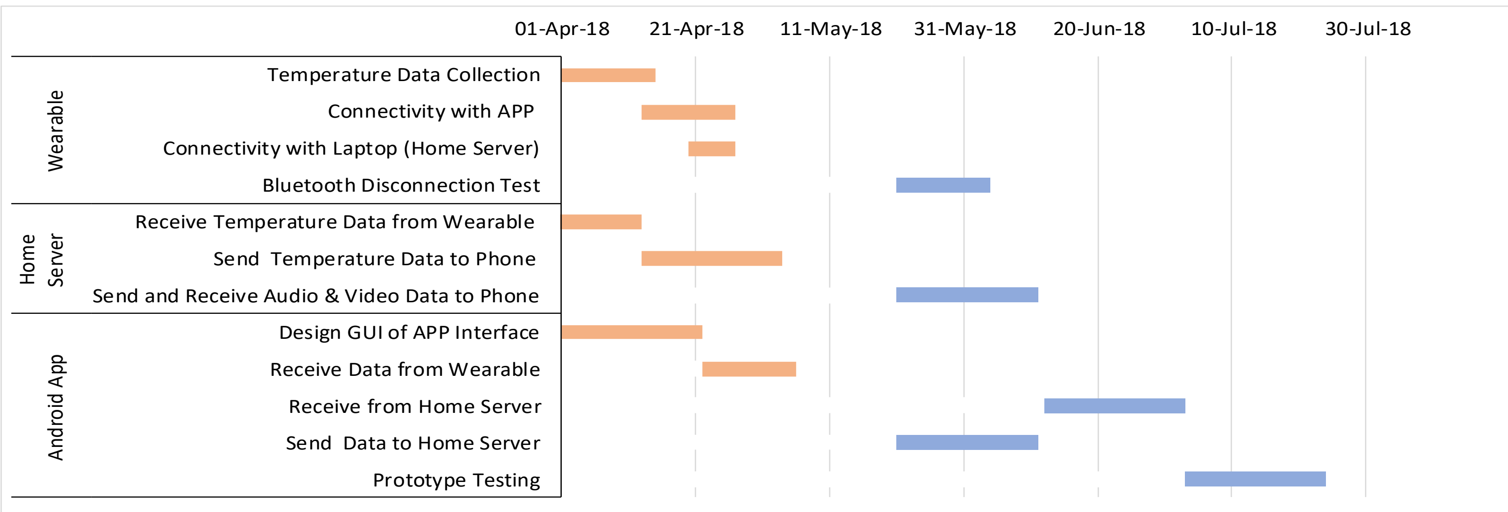


Fig. 18. the Gantt chart for our project.

## 4 CONCLUSION

We have currently met our main MDR goals, but still have some important optimizations to work on. Currently, we have a working Android app and home server which successfully connect with wearable devices to extract data. An alarm based on connectivity loss has yet to be designed for the Android device. The Android app can send control signals to the wearable device (LED ON/OFF).

### Appendix

Application Engineering

Rigorous testing of the Bluetooth range has yielded interesting results. The connectivity is up to 20 meters without any interference. With interference the connectivity range drops drastically below 10 meters.

TABLE 3 testing of the Bluetooth range

| Distance(mtr) | Bluetooth Strength | Status |
|---|---|---|
| 21 | -- | No |
| 20 | -79dBm | connected |
| 15 | -68dBm | connected |
| 10 | -50dBm | connected |
| 5 | -45dBm | connected |
| 1 | -42 bBm | connected |

The 220mAH CR2032 coin cell battery lasts up to 30 days on standby, and will continuously send streaming sensor data to a connected device for up to 7 days.

TABLE 4 Cost

| Item | Estimated Cost |
|---|---|
| Bluetooth Dongles | ￥25.5 |
| Video Sensor | ￥25 |
| Audio Sensor | ￥50 |
| Coin Cell Battery | ￥13 |
| Wearable Devices (MetaWear C) | ￥380 (US$60) |
| Laptop | ￥3000 (Testing Purpose) |
| Android Device | ￥2000 (Testing Purpose) |
| TOTAL COST | ￥493.5 (US$77.5) |

## ACKNOWLEDGMENT

Team BabyGuard would like to thank Mr. Huron and the University of Massachusetts Amherst for funding and supplying equipment for our project. We would also like to thank Professor Russell Tessier for guiding us throughout the project phase and recommending sensors and speaking with us about our project through regular meetings and follow-ups.

## REFERENCES

[1] http://www.dailymail.co.uk/news/article-2989404/Child-snatchers-abducting-200-000-boys-girls-openly-selling-online-10-000-kidnapping-epidemic-destroying-families-China.html

[2] http://abcnews.go.com/US/missing-children-america-unsolved-cases/story?id=19126967

[3] https://www.cdc.gov/safechild/burns/index.html

[4] https://www.safekids.org/heatstroke

[5] https://www.banggood.com/Vvcare-851-3_5-Inch-2_4GHz-Wireless-Baby-Monitor-TFT-LCD-Video-Night-Vision-2-way-Audio-Infant-Baby-p-1256151.html?rmmds=search&cur_warehouse=CN

[6] https://www.banggood.com/Digoo-DG-KF30-Mini-Smart-Finder-Wireless-Bluetooth-Alarm-Anti-Lost-Locator-Kid-Key-Phone-Tracker-p-1241224.html?rmmds=search&cur_warehouse=CN

[7] https://mbientlab.com/documents/MetaWearCPSv0.5.pdf

[8] https://mbientlab.com/documentation/

[9] https://mbientlab.com/developers/

[10] https://mbientlab.com/tutorials/