Project Write-Up GetFit

System Explanation

The GetFit wearable wireless fitness tracker is everyone's next best friend. Our system uses the beaglebone as the brain of the fitness tracker. Our system is wireless and independent of the host because the fitness tracker runs at boot using systemD. The system is connected to Wifi via a wifi module. The key features of the fitness tracker are laid out below.

- **A heart rate monitor** to measure your BPM through a sensor connected to your ear or finger tip. Your heart rate is read through the sensor, converted into beats per minute (BPM), and updated every 15 seconds. This feature is connected to the beaglebone via GPIO, and runs on boot in its own thread. The heart rate thread runs continuously.
- **A pedometer** to track your steps via an accelerometer that calculates acceleration on 3 axes while moving. The accelerometer is connected to the beaglebone via I2C, and runs on boot in its own thread. The accelerometer thread runs continuously.
- **A timer mechanism** controlled via buttons and the boystick. The timer is set by moving the joystick left. Minutes and seconds are selected via a button, and changed with the joystick moving up and down. The timer is then started by moving the joystick left again. The timer can be prematurely canceled using another button. The buttons and joystick are connected to the beaglebone via GPIO, and the timer runs on its own thread. The timer thread is only active when the user sets the timer by pushing the joystick left. After the timer is finished (or canceled), the timer thread is stopped.
- **An OLED display** to display to the user the different modes of the fitness tracker. The user can select which feature they would like to see, such as their heart rate with a heart actually beating on the screen, the total steps taken, and the timer. When the user selects the timer, there are instructions displayed on the OLED for how to set the timer and start it. The OLED is connected to the beaglebone via I2C.
- **A web server** hosted on the beaglebone to select different modes of the fitness tracker and display the same features you would see on the board via the OLED display. The web server allows you to keep track of your progress and statistics. This is all accomplished using a UDP socket to communicate with the beaglebone, and send data across the socket to be displayed on the OLED display.
- **Voice recognition** via text-to-speech that allows you to select the different modes of the fitness tracker, such as heart rate, steps or timer. This feature is not implemented wirelessly as we were unable to load the necessary libraries on the beaglebone due to lack of memory. If we had more time, we could have reflashed the board to allow us to implement the speech-to-text on the beaglebone This feature works when the beaglebone is connected via micro-usb to the host.
- **Wireless server access** as long as you're on the same network as the beaglebone, its web server can be accessed from any device. Using a wifi module, the beaglebone can be connected to any network wirelessly, allowing the flexibility of viewing the server anywhere, as long as there's a wifi network available.

System Diagrams

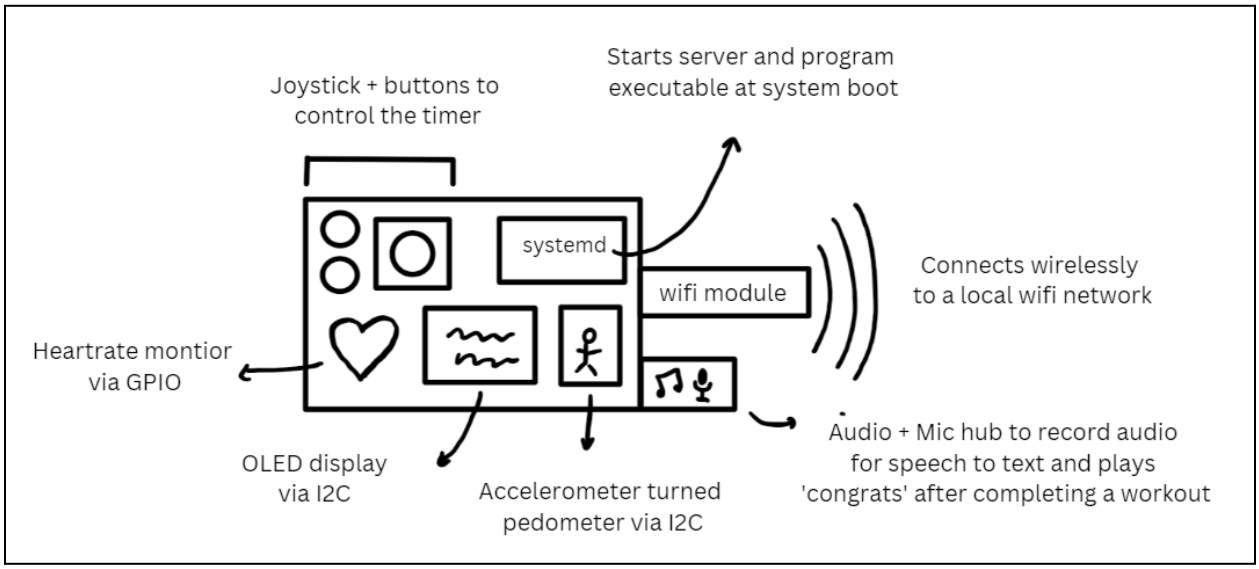

Figure 1: BeagleBone components

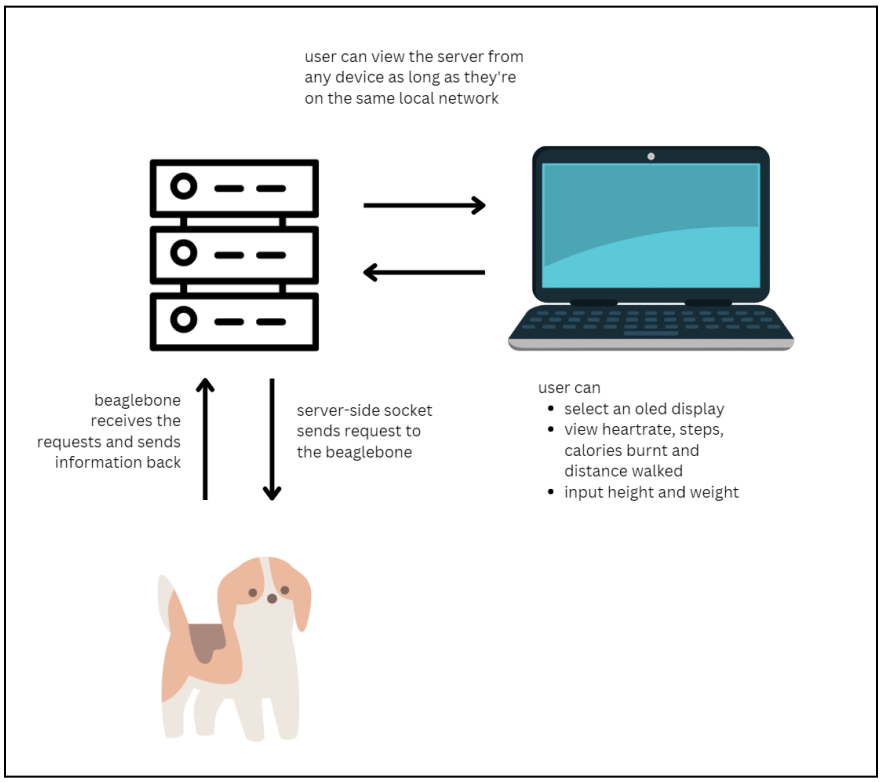

Figure 2: Web-Server Communication

Features Table

Description	Host/Target	Completeness	Code	Author	Notes
Heart Rate Monitor	T	5	C	Samin	Generates BPM
Websocket Server	T	4	JavaScript, HTML, CSS, C	Sarah, Brian's UDP server example, C server example	The server communicates to the BBG via a UDP socket to display the user's stats (steps, BPM, etc.), control the trackers display, and input the user's height and weight.
Accelerometer	T	5	C	Hazelle	Used as pedometer for step counting
Speech-to-Text	H	3	Python	Hazelle	Works on the host environment not on target due to insufficient storage.
OLED Display	T	5	C	Sarah, OLED driver found on github.	Used git library. Displays to the user the different interfaces for the different modes they can select such as steps, heart rate, setting the timer etc.
Timer	T	5	C	Samin	Able to set the minutes and seconds for the timer, plays a congrats audio when the timer is done.
Joystick	T	5	C	Samin	Used to increase and decrease the timer/ press to the right to start the timer. (Used the code I wrote for the assignment as the base)
Buttons	T	5	C	Samin/Rose	Used the code we wrote for the assignment as the base and made changes to it. The gray button has 2 modes for seconds and minutes. Green button cancels the timer.
SystemD	T	5	systemd	Rose	Based on Brian's example. Allows the project executable to run at boot.
Wifi Module	T	4	~	Rose	Able to connect to home wifi, issues connecting to SFU wifi.

Extra Hardware & Software Used

Extra hardware used in this project is as follows:

- Heart rate monitor: model Grove earl-clip heart rate sensor
- Accelerometer: model ADXL345
- Joystick (given in course hardware kit)
- 2 buttons (given in course hardware kit)
- OLED display: model 0.96 Inch OLED Module 12864 128x64 SSD1306 Driver I2C
- USB wifi module: (Brian's hardware) model USB wifi (802.11b/g/n) module
- Audio adapter (given in course hardware kit)

Extra software used in this project is as follows:

- Speech recognition
    - Audiosegment + NoiseReduce - for filtering the data from the beaglebone to remove noise, clean audio, and make the audio louder
    - SpeechRecognition + Google Cloud Storage - for transcribing wav files

- OLED driver
    - Used to handle I2C communication
    - Set up predefined display functions so that writing to the display became a matter of simple string manipulation

Video Demo:
Link to Wired Video - speech to text implemented https://youtu.be/HhFkicLCLAI
Link to Wireless Video - emailed to Dr. Brian