# Function Specification

This file includes:

Author: Huy Vu
Date: 16$^{th}$ Mar, 2017
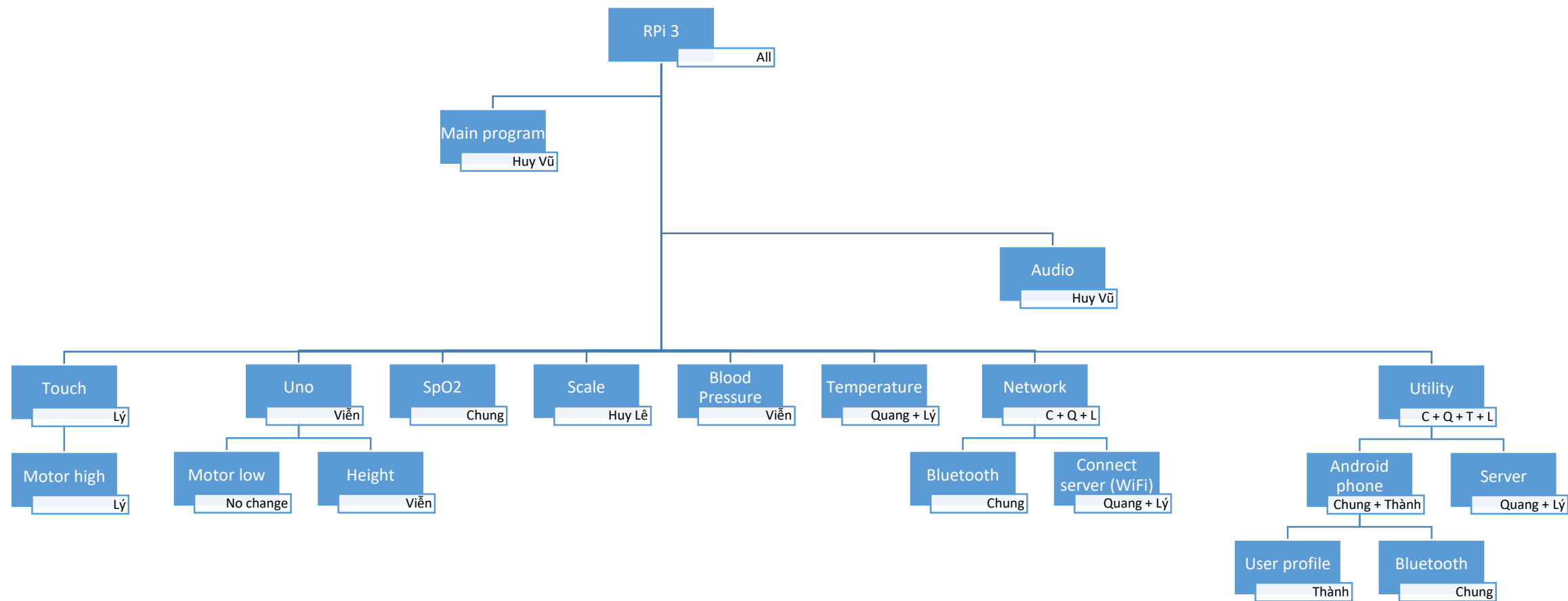
1. General rules for coding
2. Defines main flow
3. Defines desired features, desired output for each module
4. Module code

# Design & Assigned Task

# General rules

- Function name:
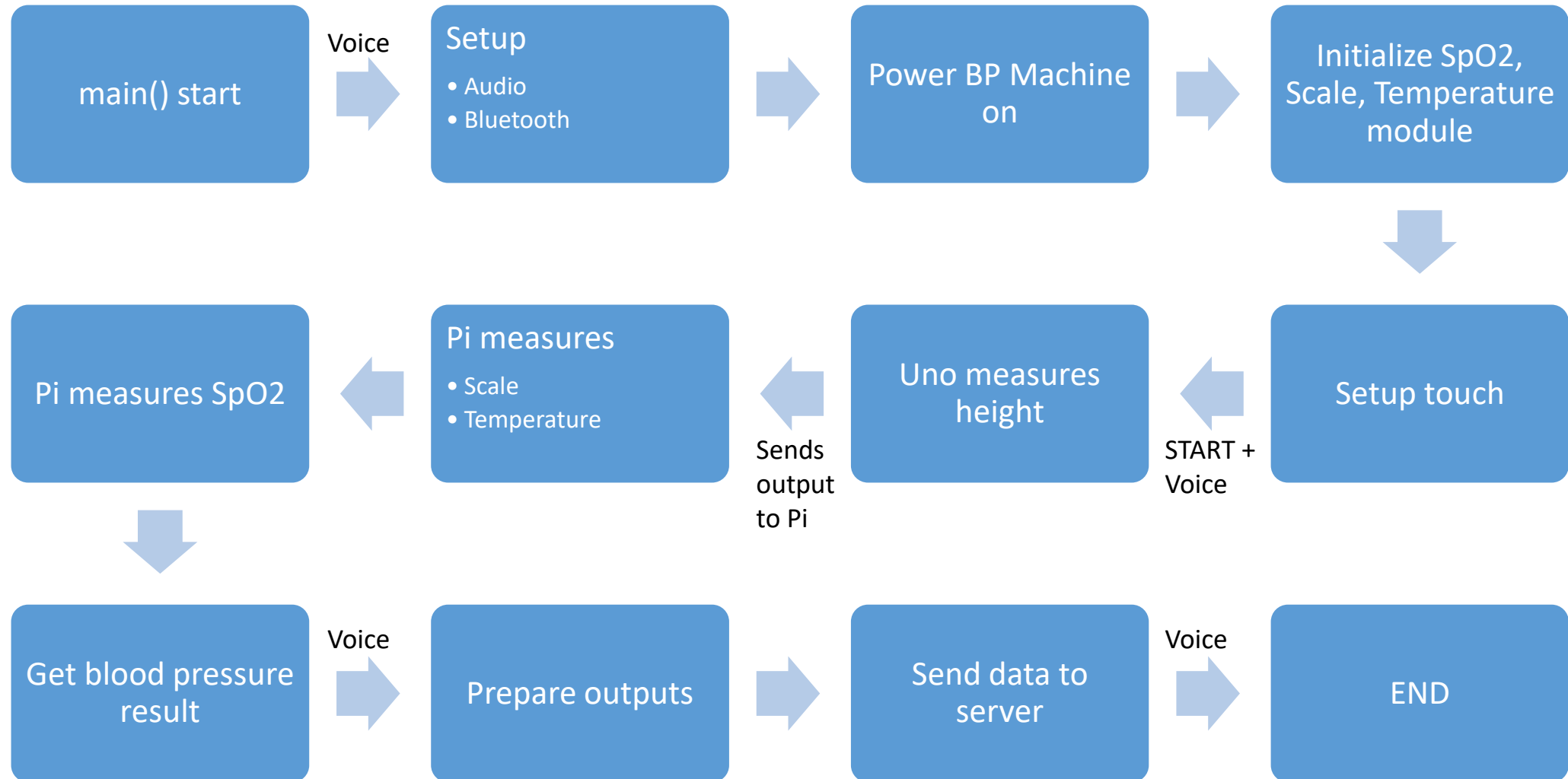  - &lt;module-code&gt;_&lt;functionname&gt;(arg1, arg2...)
    - The name should be meaningful
    - The module name is lower case
    - Ex: spo2_init(); scale_calibration(); ...
- Variable name:
  - Meaningful
- Remember to comment in source code

# General rules

- Main program will be written in Python, reasons:
  - Add new module easily
  - Raspbian and most unix distros support python natively
  - Merge whole project easily
- Other modules can be written in any language, the output of each module is an executable file and some file used for communication
- inter-process communication: via files

# Main flow

```
main() start  --Voice-->  Setup
                          • Audio
                          • Bluetooth
              -->  Power BP Machine on
              -->  Initialize SpO2, Scale, Temperature module
              -->  Setup touch

Setup touch  --START + Voice-->  Uno measures height
             --Sends output to Pi-->  Pi measures
                                      • Scale
                                      • Temperature
             -->  Pi measures SpO2
             -->  Get blood pressure result

Get blood pressure result  --Voice-->  Prepare outputs
                           -->  Send data to server
                           --Voice-->  END
```

# Main

- Huy Vu

- Description:
  - Initialize Pi
  - Call modules according to system flow
  - Handling inter-process communication

- Output:
  - Program: main.py, main_class.py (header file)

# Audio

- Huy Vu >> This module will be implemented in python

- Description:
  - Input: some mp3 files
  - Initialize audio, then generate audio_finish
  - Play mp3 according to current step.

- Output:
  - Program: audio
  - File: audio_finish

# Bluetooth

- Chung

- Description:
  - Pair Pi and the connecting Android phone
  - Get android phone mac address, store in bluetooth_result
  - When finish, generate bluetooth_finish

- Output

# Touch

- Lý
- Description:
  - Initialize and control the hand-placing platform
  - After initialize, generate touch_finish
- Output:
  - Program: touch
  - File: touch_finish

# Uno

- Viễn

- Description:
  - Initialize then wait for a file name touch_finish (uno_start), then send START signal to Uno.
  - Uno will drive the motor, measure height, then send back to Pi
  - Pi store that result in a file name height_result, generate a file name uno_finish

- Output:
  - Program: uno
  - File: height_result, uno_finish

# Scale

- Huy Le
- Description:
  - Initialize then wait for a file name uno_finish
  - Measure weight
  - Store result in scale_result, generate scale_finish
- Output:
  - Program: scale
  - File: scale_result, scale_finish

# Temperature

- Description:
  - Initialize then wait for a file name uno_finish
  - Measure temperature
  - Store result in temp_result, generate temp_finish
- Output:
  - Program: temp
  - File: temp_result, temp_finish

# SpO2

- Chung

- Description:
  - Initialize then wait for a file name scale_finish AND temp_finish
  - Measure the SpO2
  - Store that result in file name spo2_result, generate spo2_finish

- Output:
  - Program: spo2
  - File: spo2_result, spo2_finish

# Blood pressure

- Viễn
- Description:
  - Has two program: bpressure_poweron, bpressure
  - bpressure_poweron will power on the bpressure machine, initialize the auto-mechanical
  - after that, generate file bpressure_poweron_finish
  - bpressure will measure when spo2_finish exists
  - Store result in bpressure_result, generate bpressure_finish
- Output:
  - Program: bpressure_poweron, bpressure
  - Output: bpressure_result, bpressure_finish

# Pi Connect server via WiFi

- Quang + Lý
- Description:
  - Send data to server when wifi_start exists
  - Generate file wifi_finish
- Output:
  - Program: wifi
  - File: wifi_finish

# Utilities

| Server |
|---|
| • Handle result from Pi |
| • Store user data (ID, name, mac addr…) |
| • Store user result (height, weight, spo2, blood pressure, temperature) |
| • Quang + Ly |

| Android app |
|---|
| • Bluetooth<br> • See above Bluetooth slide<br> • Chung<br>• User profile<br> • Ask mentor & server team for more information ☺)<br> • Thành |

# Module code (for implementing)

| Module | Code |
|---|---|
| Audio | audio |
| Bluetooth | bluetooth |
| Blood pressure | bpressure |
| Scale | scale |
| SpO2 | spo2 |
| Temperature | temp |
| Touch | touch |
| Uno | uno |
| Connect server (wifi) | wifi |

# Summary

| Member | Task |
|--------|------|
| Huy Vũ | Main program, Audio |
| Huy Lê | Scale |
| Chung | SpO2, Bluetooth |
| Thành | User profile |
| Viễn | Blood Pressure, Uno |
| Lý | Server, Connect server via wifi (Pi), temperature, touch |
| Quang | Server, Connect server via wifi (Pi), temperature, touch |

This file can be updated by any member's request. This is not final version.

# END

All corrections, suggestions, contributions are welcome