Meobot

Author: Haohan Shi, Olivia Xu, Yanying Zhu

Electrical and Computer Engineering, Carnegie Mellon University

***Abstract*—Meobot is a desktop companion capable of moving autonomously and interacting with user through audio, video and movement. It's a helpful friend that can understand and execute simple voice commands. Subsystems include voice detection, hotword detection, facial recognition, and robot control system.**

***Index Terms*—API, control system, face recognition, robot, voice recognition**

I. Introduction

OUR robot Meobot, is a desktop companion that can interact with human. Mebot can move freely on the desktop without hitting obstacles or falling. When Meobot is called, it can recognize the voice of the caller and respond if the user ask for services. For example, Meobot would be able to display weather forecast if the user ask for weather information. The idea of making a desktop companion comes from ANKI's Vector robot, which is a home robot developed with interactive AI mechanism. Similar to Vector, Meobot is able to perform interactive behavior with user by remembering user's images and voices. Meobot's AI feature in addition to control system differentiate it from regular robots. The goal for meobot is firstly, that it cannot fall in any cases. Hitting small obstacles is allowed, while hitting large obstacles which would result from damaging the robot is not allowed. Secondly, detecting and responding to stored human voice input should have a 80% above rate of accuracy. Thirdly, detecting and recognizing stored human face from camera input should have a 90% above success rate. Finally, the response time for each command should be less than 3 seconds.

II. Design Requirements

The whole robot itself should be within 5 inch x 5 inch x 4 inch box, so that it is small enough to be as a desktop companion.

Battery time greater than 30 minutes. Batteries are separate for the robot base and RPi. The base uses four AAA batteries and the Raspberry Pi is powered by a portable charger of at least 2200mah.

Hotword detection will have 80% "will respond when you say Hi-Meo" to eliminate activation upon non-keyword, and the "false alarm" should reduce to around 1 time in 3-4 minutes under noisy environment.

Our LCD video output is expected to provide simple expressions (blinking, smiling, confusion, sleepiness), "cut-scenes" ("connecting to internet"), and info display in combination with audio output if necessary (digital clock, graphic weather indication, numbers, words).

If a command requires internet requests, it should respond within 3 seconds, otherwise it's a close to immediate response.

Obstacle detection and edge detection are expected to perform at 100%, it will always detect the obstacle and will never fall of an edge. Obstacles do not include scraps that can be run over.

Our Raspberry Pi and robot kit should successfully communicate through serial connection with zero miscommunication. Serial communication should be fast and concise and the robot should process the command within 0.5s delay. This metric is crucial for obstacle and edge detection.
Our facial recognition should be able to detect the correct person within 2 seconds of delay, and achieve at least 80% accuracy.

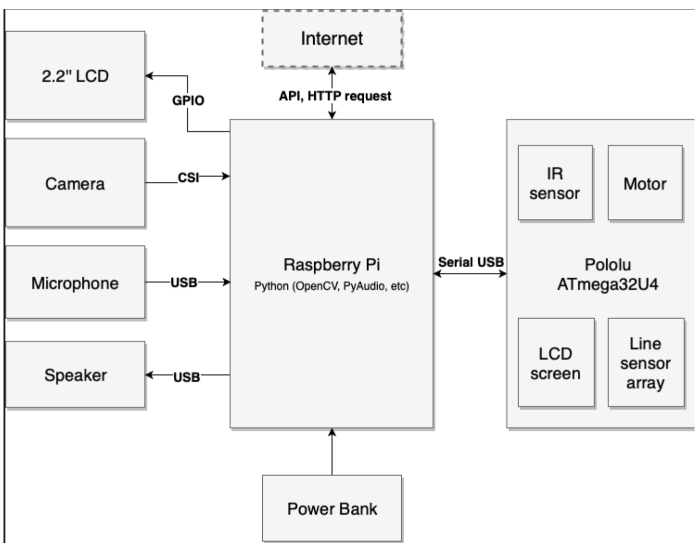
Our command recognition rate should achieve at least 80% under supported commands. This is the most important requirements for our project and is directly related to user experience of our project. This will be measured by extensive user testing under different surrounding environments.

For our robot's movement and obstacle/edge detection sensors we have chosen to use a robot kit from Pololu, so we could focus on implementing our software features. This pre-built robot, Zumo, comes with two side proximity sensors, two front IR proximity sensors, three line sensor arrays on the bottom that also act as edge detection. The sensors give prompt feedback, but the major challenge is the proximity sensors only work within a short distance.

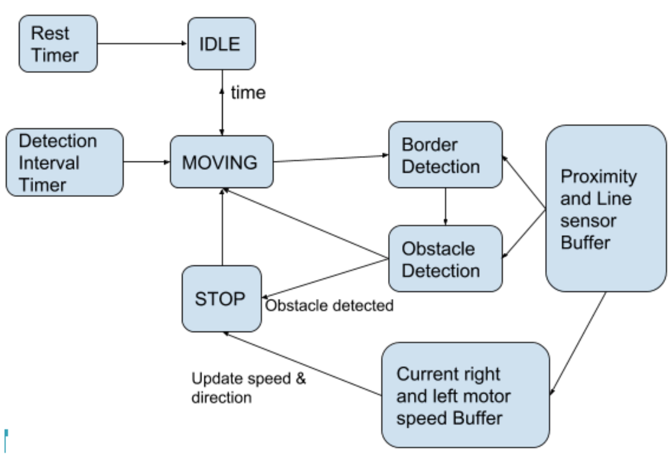
Meobot requires that there are constraints and limitations for its environment. Meobot is meant for a relatively clear desktop or counter space without complex obstacles that require path-planning. Small scraps like shreds of paper are acceptable and will be ignored. It is expected to wander in this given space while avoiding running into obstacles and falling over edges. Also, this robot requires a relatively quiet surrounding so that all the microphone capture and hotword detection will work properly.

III. Architecture and/or Principle of Operation

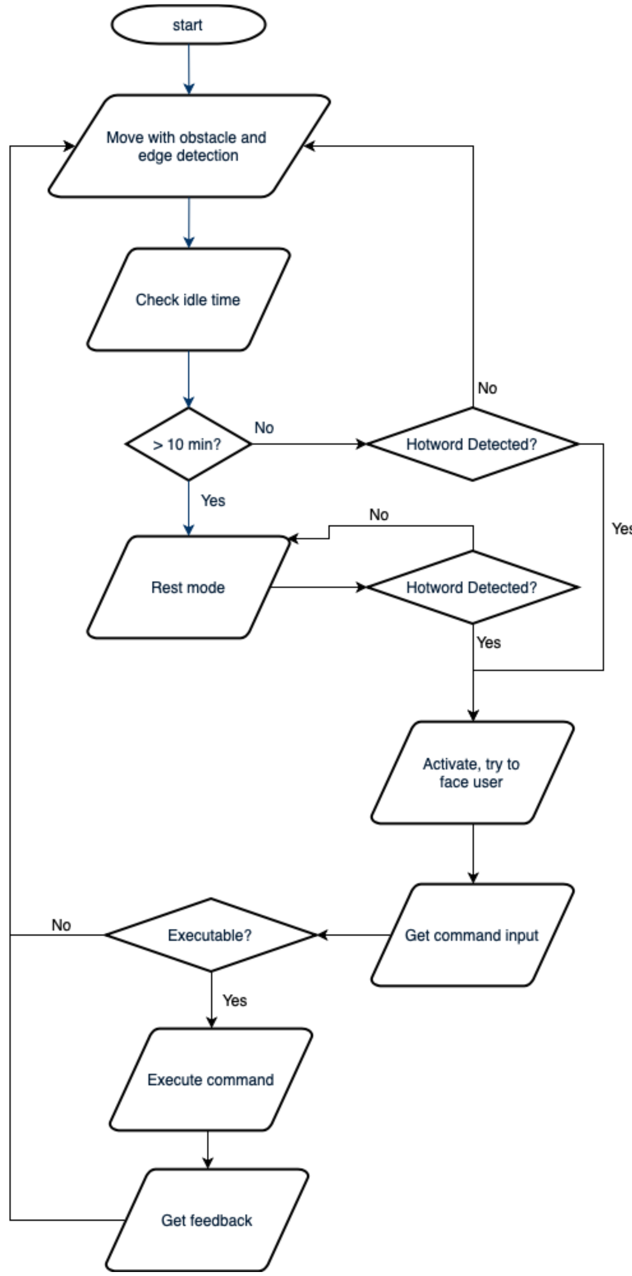
Our entire system uses a Raspberry Pi as a central control unit. Movement sensors are all on the Pololu Robot kit. Our Raspberry Pi will take in information from proximity sensors for object/edge detection, camera for video input to combine with proximity sensors to implement a "follow human" feature; the Raspberry Pi will also communicate with API and HTTP requests to get information; our robots outputs are the microphone for audio answers and responses that will complement what's displayed on the LCD.



(a)



(b)



(c)

(a) Overall system
(b) Self-controlled Movement System
(c) Voice activation & command flow

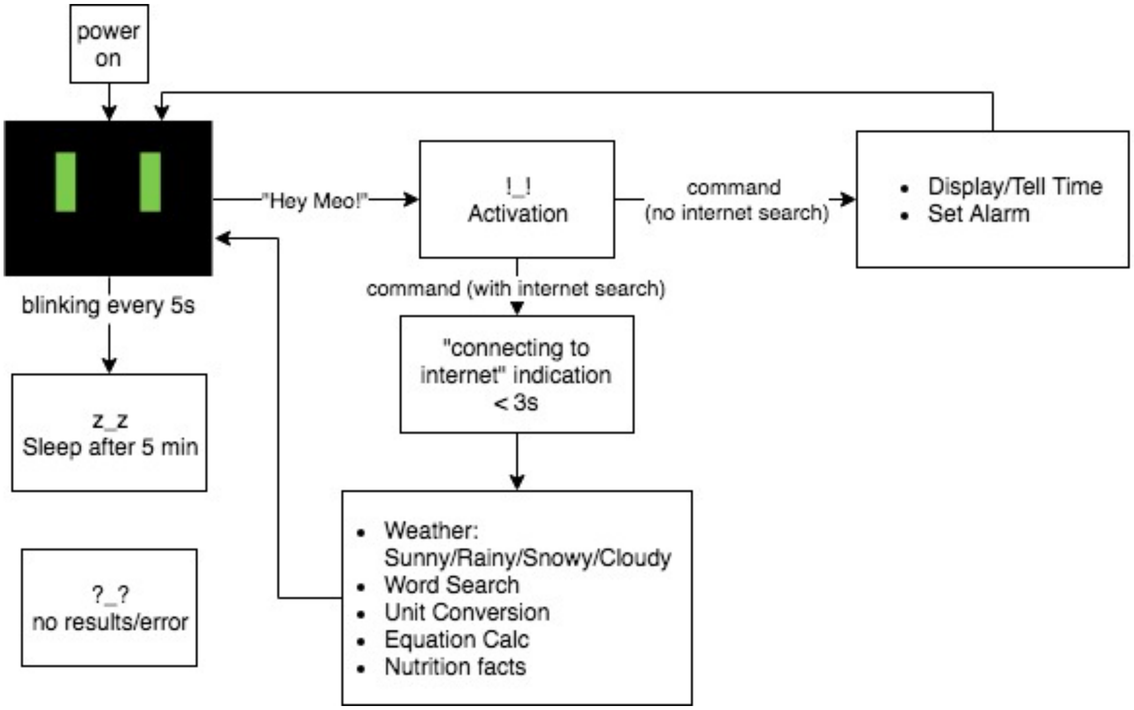


Fig. 1. LCD visual display flow chart

IV. Design Trade Studies

A. Choice of Communication Hub

We choose Raspberry Pi 3B+ as our ideal communication center. The most important reason is it has enough computing power and expansion interfaces such as CSI camera interface, 4 USB interface, and on-board WiFi. In addition, the GPIO ports enable us to connect our LCD display without worrying too much about driver and power. By using Raspberry Pi, we can successfully separate computation part and movement control part and make the whole project dividable.

B. Choice of Robot Platform

In our project, we choose to purchase a commercial robot platform to develop on instead of building our own robot from scratch for two main reasons.

First, our area specializations are mainly software and hardware system, instead of circuit and signal processing, so we don't want to spend too much time on researching how to make something movable. Our main focus should be the software features and different capabilities of our helpful and cute robot.

Second, the commercial robot kit usually comes with pre-built library and example code so that we can directly learn and implement on the robot. Also, these robot kit usually comes with on-board sensors and extension capabilities so that they will significantly reduce our cost and time in the future.

At the very beginning, we think the Pololu m3pi Robot + mbed NXP LPC1768 Development Board Combo should be good for our project because it has extra GPIO ports for us to connect all the additional sensors we need such as proximity sensors. But after our discussion we found this kit has several issues.

First, it is a two-wheel robot kit, which will tend to be fairly unstable after putting a lot of load on top of it, especially due to the fact that we may add power bank to power the raspberry pi and attach camera on the front.

Second, the development environment on this robot kit is unfriendly to group project, this robot kit needs to be coded in a web mbed IDE instead of local one. This may cause a lot of difficulties to our group development and debugging.

Finally we decided to use Pololu Zumo 32U4 Robot, and we finally ordered this one for several reasons. First it has built in IR proximity sensor system on the front and two sides of the robot, which is exactly what we need for obstacle and edge detection. Second, it uses a built-in Arduino-compatible ATmega32U4 microcontroller, which means we can write in Arduino IDE using its library. Third, this robot uses two tracks which solves the stability problem.

C. Choice of Microphone and Speaker

We decide to use SunFounder USB 2.0 Mini Microphone and HONKYOB USB Mini Speaker for the following reasons. First, they are all using USB driver-free interface. Which is great for Raspberry Pi because we are using our GPIO ports for LCD display. Driver-free means we only need to do some basic setup for them to work on Raspberry Pi without writing code or change lots of system settings.

Second and the most important reason is they are fairly small with will fit our size specification. The USB microphone is 0.87 inch x 0.71 inch x 0.28 inch and the USB speaker is 3.14 inch x 1.2 inch x 1.78 inch. Which will not take too much space when attaching to the Raspberry Pi.

D. Choice of Camera

We are using Raspberry Pi Camera Module V2. The first reason is that it is official certified camera, which means it can directly plug into Raspberry Pi's CSI interface and use. Second, it has fixed focus lens, which is suitable in our use case because users are always within roughly 1 meter range from the robot, and this eliminates the time to adjust the focus and significantly decrease our process and response time. Third, it has high quality 8 megapixel Sony IMX219 image sensor custom designed add-on board for Raspberry Pi, which is good for mounting purpose.

E. Choice of LCD Display

We are using 2.2'' 18-bit color TFT LCD display from Adafruit for the following reasons.

First is its size, 2.2'' is ideal for our project because it is a little narrower than our Pololu robot kit base, which is great for aesthetic purpose.

Second, this is a product from Adafruit, and Adafruit has ILI9341 Python library for us to use directly. Which will make it easy for us to draw characters, pixels and even simple animation on this display.

Third, it is compatible with Raspberry Pi's GPIO ports, by connecting the ports with correct GPIO port, it is good to go. This eliminates the need for extra electrical components or power input.

V. System Description

A. Hotword Detection

We are using Snowboy hotword detection on Raspberry Pi. It is embedded and light-weighted, and does not require any Internet connection. This frees up a lot of computation resources and protects user's privacy. Based on our hotword detection requirements and specifications, we will adjust the input gain of the microphone and sensitivity of the detection model so that we can still achieve a high detection rate without lots of "false alarms".

B. Communication Mechanism

We are using serial communication between Raspberry Pi and Pololu robot kit. We are designing a serial communication protocol so that the Raspberry Pi will act like a commander and Pololu as responder. The sample communication code are like follows:

Sample Raspberry Pi code:

```python
import serial
ser = serial.Serial('/dev/ttyACM0', 9600)

while(1):
    userInput = input("Send: ")
    ser.write(userInput.encode())

    response = ser.readline()
    print(response.decode())

```

Sample robot code:

```cpp
void loop()
{
  if (Serial.available() > 0) {
    a = Serial.readString();// read the incoming data as string

    lcd.clear();
    lcd.print(a);
    // say what you received:
    String response = "I received: " + a;
    Serial.println(response);
    if (a == "forward") {
      motors.setSpeeds(100,100);
    }
    if (a == "stop") {
      motors.setSpeeds(0,0);
    }
    if (a == "back") {
      motors.setSpeeds(-100,-100);
    }
  }
}
```

C. Movement Control

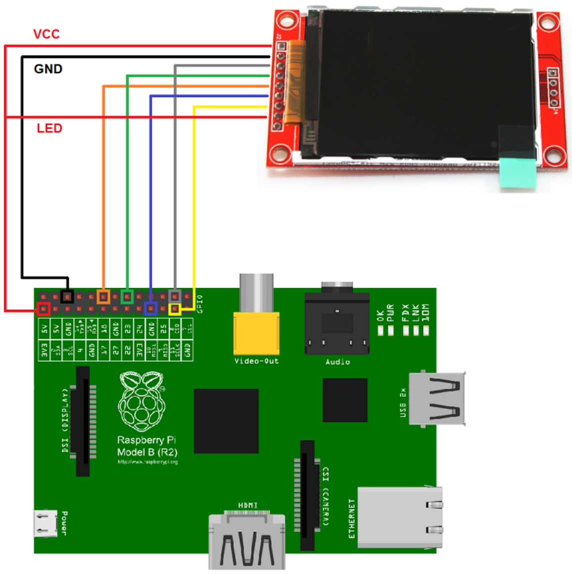
Meobot has a self-controlled movement system that allows it to make moves by itself when there is no command given. Meobot would either move around on the table freely or stay still. The behaviour of whether to stay still or move is randomly chosen at different times. When Meobot moves, it would constantly checking the environment using the three proximity sensor which are located at left, right and front. When Meobot detect obstacles, it would stop right away to recalculate directions, attempting to avoid hitting any obstacles. After that meobot would resume to moving state with the updated motor speed.

D. Facial Recognition

We are using Python openCV library for facial recognition. The whole process consists of three steps. Data gathering, training and recognition. We are using our own photos taken by the Raspberry Pi camera, around 30 photos for each person. For training, we are going to use openCV's built-in Local Binary Patterns Histograms face recognizer. By using our training model, we are going to "predict" the category when given a image, and output the possibility that the image belongs to specific training data.

E. LCD Display

For our visual output, we are using a 2.2'' LCD display from Adafruit. We will use Adafruit's ILI9341 Python library, from this library, we will build another layer of interface to correctly show English letters, numbers, weather information and some simple animation.



Source:
http://raspberrypi4u.blogspot.com/2014/12/raspberry-pi-connect-tft-lcd-with.html

VI. Project Management

A. Schedule

See end page

B. Team Member Responsibilities

Member	Task
Haohan Shi	Voice/Hotword detection and voice command processing Facial recognition
Olivia Xu	LCD driver and display development Control - Border detection and Testing
Yanying Zhu	Control - Obstacle Detection and Path planning Control System Integration and Testing
Team	System Integration Appearance Design and Implementation User Testing

C. Budget

Item	Quantity	Price
Raspberry Pi 3 B+	2*	$38
Pololu Zumo Robot	1	$150
Breadboard Jumper Wires Kit	1	$6
Raspberry Pi Camera v2	1	$17
2.2" 18-bit color TFT LCD display	1	$25
Mini USB Microphone	1	$6
Mini USB Speaker	1	$12
Total		$292

*Our first purchased Raspberry Pi stopped working so we ordered another one.

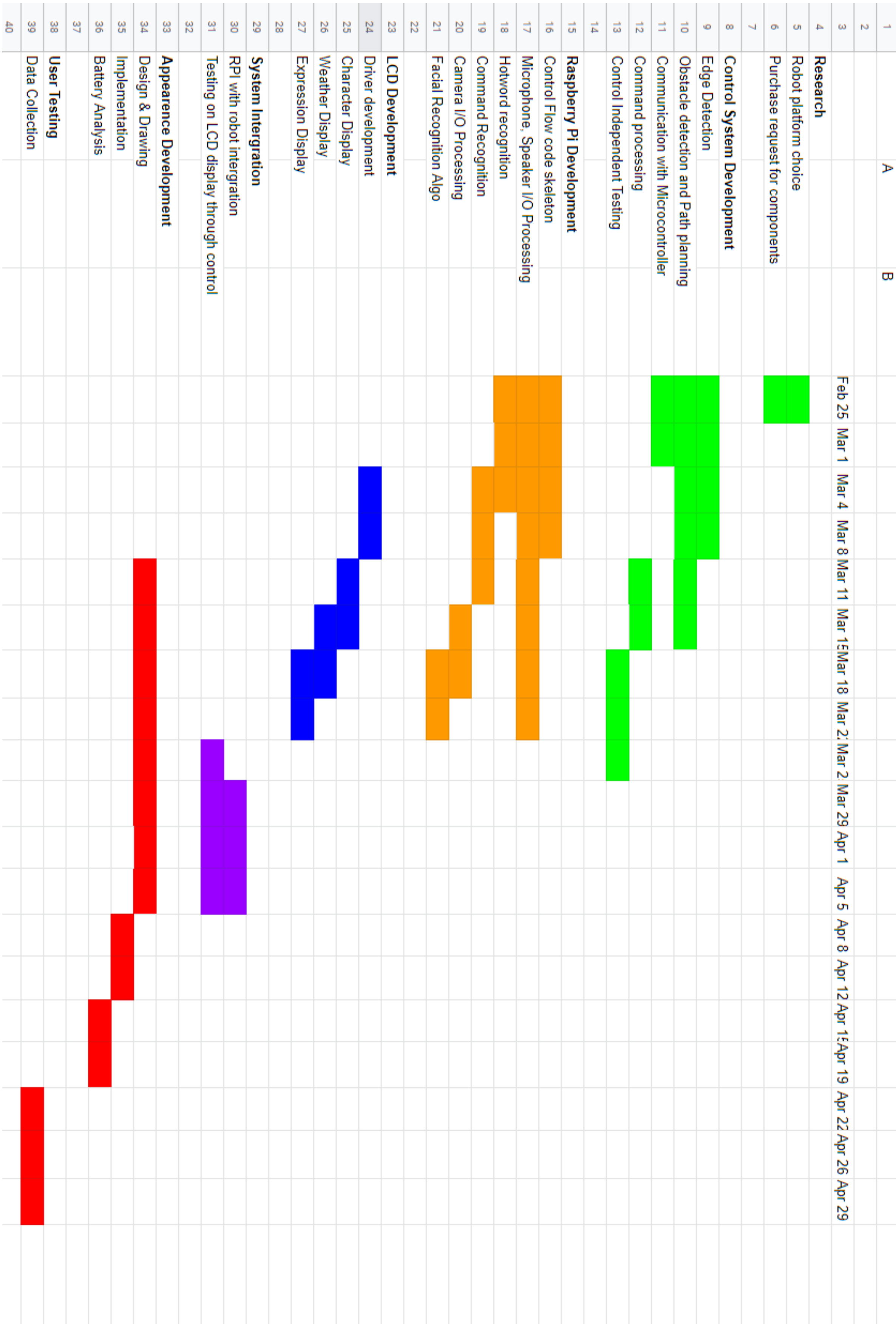
D. Risk Management

We have identified several risk factors during the design and sub-system implementation and system integration stage.

- Difficulty in integrating proximity sensors (which already has a "follow close target" function in the Pololu library) and face recognition to implement "turn to follow human"
- Difficulty in allocating Raspberry Pi processing power for smooth operation
- Difficulty in getting relatively fast internet (API, HTTP requests)
- Insufficient time for integration testing
- End up with very limited functionalities

VII. Related Work

http://raspberrypi4u.blogspot.com/2014/12/raspberry-pi-connect-tft-lcd-with.html
https://forum.pololu.com/t/mount-raspberry-pi-on-zumo-32u4/8019
https://learn.adafruit.com/user-space-spi-tft-python-library-ili9341-2-8?view=all
https://www.pololu.com/blog/577/building-a-raspberry-pi-robot-with-the-a-star-32u4-robot-controller


A
B
Feb 25
Mar 1
Mar 4
Mar 8
Mar 11
Mar 15
Mar 18
Mar 2
Mar 2
Mar 29
Apr 1
Apr 5
Apr 8
Apr 12
Apr 15
Apr 19
Apr 22
Apr 26
Apr 29
Research
Robot platform choice
Purchase request for components
Control System Development
Edge Detection
Obstacle detection and Path planning
Communication with Microcontroller
Command processing
Control Independent Testing
Raspberry Pi Development
Control Flow code skeleton
Microphone, Speaker I/O Processing
Hotword recognition
Command Recognition
Camera I/O Processing
Facial Recognition Algo
LCD Development
Driver development
Character Display
Weather Display
Expression Display
System Intergration
RPI with robot intergration
Testing on LCD display through control
Appearence Development
Design & Drawing
Implementation
Battery Analysis
User Testing
Data Collection