Peter Mohr

Individual Work package

Group: Shocking Engineers

As the sole computer science major in the group, Shocking Engineers, the tasks below pertain to the software components of the product. However, it should be noted that all group members will likely contribute to each step as their expertise and time allows.

The tasks below are in the sequence they need to be completed.

**Task 1:**

A working python environment needs to be created for code implementation between the Raspberry Pi and the MLX90640 Thermal Camera

**Necessary Resources/Code Libraries**

-Python 3.6

-Adafruit\_mlx90640.mpy

-Adafruit\_bus\_device

-Adafruit\_Blinka

**Difficulty/Time Estimate: 1 Week**

I already have experience interfacing with the camera on a Linux desktop computer and have explored tutorials completing this on a Raspberry Pi. I am familiar with the QT STEMMA connector the Pi utilizes I am familiar with using and operating a Raspberry Pi.

**Deliverable:**

At this end of this task, the data from camera should be able to be manipulated and analyzed in a standard python environment.

**Task 2:**

Second a working python environment needs to be created for code implementation between the Raspberry Pi and the VL53L0X Time of flight Sensor

**Necessary Resources/Code Libraries**

**-**adafruit-circuitpython-vl53l0x

**Difficulty/Time Estimate: 1 Week**

I am familiar with the QT STEMMA connector the Pi utilizes I am familiar operating a Raspberry Pi and interfacing with such devices.

**Deliverable:**

At this end of this task, the data from Time-of-Flight sensor should be able to be manipulated and analyzed in a standard python environment.

**Task 3:** Once data from the sensors can be integrated into the python coding environment the core algorithm for detecting a person’s temperature from the sensor data will be created.

**Necessary Resources/Code Libraries**

**-**core Python3.6 environment

**Difficulty/Time Estimate: 1 Week**

In this section I am creating the algorithm that analyzes the temperature and distance data to create a single accurate (+-0.5 Celsius) result. This step will not involve validation. The idea is to simply create an algorithm, given accurate data, that will produce a single accurate temperature measurement.

**Deliverable:**

At this end of this task, the camera will be able to visualize a person and send its data to the Raspberry Pi, then the algorithm on the Pi will return a single temperature, that in theory, reflects the persons core temperature. (Given some conditions, such as the persons forehead is visible and not covered in perspiration)

**Task 4:** Validation of algorithm

**Necessary Resources/Code Libraries**

**-**Volunteers for core temperature validation

**Difficulty/Time Estimate: 4 Weeks**

This task has the most potential for variability and project failure. This task will involve taking core temperatures of actual people via a validated method (i.e., a digital oral thermometer). Then measuring those same people with our thermal camera system. Then based on how these measurements compare, modify the algorithm created in task 3 for accuracy. Furthermore, as a group we will need to decide how many volunteers are ultimately necessary for validation. This task will likely be the most time consuming and perilous for the project.

**Deliverable:**

At this end of this task, the camera will be able to visualize a person and send its data to the Raspberry Pi, then the algorithm on the Pi will return a single temperature that is reflective of that person’s core temperature to +-0.5 Celsius. (Given some conditions, such as the persons forehead is visible and not covered in perspiration)

**Task 5:** Creation of User Interface/Temperature Display

**Necessary Resources/Code Libraries**

**-**Research on this topic needs to be completed

**Difficulty/Time Estimate: 3 Weeks**

This task involves taking the system created in Task 4 and creating a program to display the result/instructions to the end user on the LCD screen attached to the device. Examples of instructions would be, “Please step forward to have your temperature taken”, “Your temperature is 98.6 degrees, no fever detected”. “Temperature is too low please remove any hats or covering on your forehead”, “temperature 100.1, fever detected”. I have not personally worked on creating applications in python with a graphical user interface and would like to allow myself time to learn the necessary libraries.

**Deliverable:**

At this end of this task, a person should be able to use our device to have their temperature taken without any instructions.