Functional Specification

Year: 2019 Semester: Fall Team: 1 Project: IntelliFace

Creation Date: 08/21/2019 Last Modified: 08/30/2019

**Authors:**

**Member 1: Rtvik Sriram Bharadwaj** [**bharadwr@purdue.edu**](mailto:bharadwr@purdue.edu)

**Member 2: Ishaan Ahuja** [**ahujai@purdue.edu**](mailto:ahujai@purdue.edu)

**Member 3: Pratyaksh Sharma** [**sharm235@purdue.edu**](mailto:sharm235@purdue.edu)

**Member 4: Abhay Sasidharan** [**asasidh@purdue.edu**](mailto:asasidh@purdue.edu)

Assignment Evaluation:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Item** | **Score (0-5)** | **Weight** | **Points** | **Notes** |
| **Assignment-Specific Items** | | | | |
| **Functional Description** |  | x3 |  |  |
| **Theory of Operation** |  | x3 |  |  |
| **Expected Usage Case** |  | x3 |  |  |
| **Design Constraints** |  | x3 |  |  |
| **Writing-Specific Items** | | | | |
| **Spelling and Grammar** |  | x2 |  |  |
| **Formatting and Citations** |  | x1 |  |  |
| **Figures and Graphs** |  | x2 |  |  |
| **Technical Writing Style** |  | x3 |  |  |
| **Total Score** |  | | |  |

5: Excellent 4: Good 3: Acceptable 2: Poor 1: Very Poor 0: Not attempted

General Comments:

1.0 Functional Description

The IntelliFace (Intelligent Interface) provides a User Interface with a facial recognition system to authenticate individual users. It will also serve as a passive temperature warning indicator and alert the user through real time push notifications through a messaging service. The sensor data includes an ambient light sensor to adjust the monitor brightness, an indoor temperature sensor, and an IR sensor to read user gestures to interact with their UI, all data read from the STM32F0 MCU transmitted to an Nvidia Jetson Nano through an USART line.

2.0 Theory of Operation

The IntelliFace will be designed to consume a variety of sensor data. Hence, a variety of engineering principles and techniques will be utilized to read and process this input data from the different sensors and camera. The system wakes up upon interaction with the IR sensor connected to the STM32F0 MCU. The sensor uses a light sensor to detect the specific wavelength in the infrared spectrum in order to sense aspects of the surroundings, which heavily relies on the scientific principles from the world of physics.

The project uses the MIPI camera to authenticate users and this feature implements the use of computer vision principles available on a number of cloud providers, a few in consideration being AWS Rekognition, GCP Face Recognition, Azure Face, with provisions to move to a locally running neural network to identify users if time permits. Furthermore, the Jetson Nano and STM32F0 will be using USART to communicate data from the IR, Temperature and Ambient Light sensors. The theory of operation can be visualized through the functional block diagram in Appendix A.

3.0 Expected Usage Case

The IntelliFace is intended for stationary indoor usage in a living or drawing room, ideal conditions being non-humid areas with moderate temperatures, although the components are planned to be sealed to avoid any major damage in harsher environments. The information displayed on the dashboard is tailored to each user, hence necessitating the need for a facial recognition system which will be performed using a low-cost online API, such as the Azure Face Recognition or AWS Rekognition to start with, but with room for expansion into a lightweight neural network run locally on the Jetson that will contain training sets for each user. The unusual temperature alert is intended as a passive system that will communicate a real time alert to the user through a PagerDuty, Slack or Google Voice API. All users will be required to submit a minimum of two photos of themselves to train the recognition system.

4.0 Design Constraints

4.1 Computational Constraints

The Computational functions for the STM32F0 MCU are:

* Handle asynchronous temperature sensor readings with a functional timer and interrupt handler to act as a temperature anomaly detector, waking the Jetson to alert the user if need be
* Perform gesture detection and wake recognition using the IR sensor
* Send sleep signals to the Jetson upon user inactivity/sleep command as read from the IR sensor
* Adjust UI brightness on the Jetson using ambient light detection from a series of LDRs
* Transmit all of the above data to the Jetson through a USART line

The Computational functions for the Nvidia Jetson Nano so far have been listed as:

* Interface with the Jetson camera, and run necessary APIs to authenticate the user
* Run the user’s dashboard, with their individual accounts correctly synced and displayed based on the authenticated user.
* Read incoming data from the STM32F0 MCU, and perform any necessary actions including brightness adjustments, send alerts for unauthenticated user access and temperature anomalies
* (if chosen and time permits) Train the neural network (once at device initialization, with images of every user), and perform facial recognition locally instead of relying on an online API

The STM32F0 MCU is not expected to consume an excessive amount of memory as it relays live data to the Jetson through a USART, hence the usage of a storage component is unwarranted.The asynchronous tasks performed on the STM32F0 require simple ADC reads and basic computation, specifically unit conversion and basic floating point arithmetic in unit conversions and data parsing, and does not warrant the usage of a more powerful MCU.

The Jetson Nano will require a larger SD card (currently functioning on 16 GB, will require 64 GB) for the storage of user data, especially if the local neural net option is chosen to store the user data and images, as well as continuously improving the model. The board itself was built to withstand the amount of computational usage needed for Reinforcement Learning, Bayesian computations, Linear Regression models and Decision Trees, and it was thus chosen in the event that a local neural net is implemented. The extra computational power is also intended to be used for over-the-air updates through CI/CD.

4.2 Electronics Constraints

The major project components we are using for our project are Temperature Sensor (TMP36), LDR, IR Sensor (Gesture Sensor), USART. USART is the Universal Synchronous/Asynchronous Receiver/Transmitter and is used as an interface between the NVIDIA Jetson and the STM32. The USART is used for transferring of sensor data from the STM32 to the Jetson. We would be using GPIO with ADC for the TMP36, LDR, and the IR Sensor. USART requires a memory buffer that needs to be created in STM32 which would then be communicated to the Jetson.

4.3 Thermal/Power Constraints

Due to the stationary indoor use-case of this product, the Jetson board and the STM32F0 MCU will require a wall power outlet, and hence will also require appropriate stabilizers to ensure that the internal components are not damaged by power fluctuations. The Jetson has an inbuilt aluminum heat-sink, and will not require additional cooling mechanisms unless the compute demands start to test the limits of the available power, which we do not expect it to for our use case as it is built to comfortably handle operations like cryptocurrency miners and deep learning programs. The STM32F0 will handle passive asynchronous tasks like ADC reads which will fall into its operating temperature margin on a long term run at 73℉ (room temperature), as referred to from the table below, all data obtained from the respective controller’s official datasheets.

|  |  |  |  |
| --- | --- | --- | --- |
| MCU | Operation | Power | Temperature at operation |
| STM32F0 | Passive Mode (Async. ADC) | 0.25W | -40℉ - 180℉ |
| Run Mode per MHz | 0.75W |
| Jetson Nano | Rest Mode | 0.5W | -40℉ - 150℉ |
| Power efficiency mode per MHz | 5W |
| High performance mode per MHz | 10W |

4.4 Mechanical Constraints

As the device is a stationary wall-mounted/tabletop mirror, the mechanical restrictions are all humidity/heat based. It’s potential for usage in bathrooms will necessitate the water-proofing of all materials, while leaving certain key sensors (in particular, the temperature and the light sensors) exposed to prevent data aberrations. Shock-proofing the device is not required as it is stationary and is not meant to be tampered with. We are aiming for the overall mirror dimensions of 48” x 32”, and the actual hidden monitor dimensions at 24”, and for optimal operations, will need to be kept in a fog proof environment. The weight of the mirror will be a factor for wall-mounting, at an expected total weight of 15-20 pounds including the wooden frame, mirror face and the monitor.

4.5 Economic Constraints

Competitor products on the market list for prices ranging from $500-1000 (from [6], [7]) depending on the number of features, a majority of which rely on Amazon Alexa and Google Home APIs. To minimize the costs, mass ordering PCBs and displays would drive the costs down to a total of $250-300 for our implementation, and by selling at $450, we can gain revenue with an operable profit margin. Maintaining the mirror interface through over-the-air updates is simple as the mirror is connected to Wi-Fi, and the project being completely open-sourced would allow simple continuous integration and deployment as and when changes are made, a feature that most other mirrors on the market do not support. Adding newer features is also made simple as the Jetson Nano can power more tasks than what is currently performed, future features being a virtual fitness coach and integration with fitness trackers.

4.6 Other Constraints

Facial recognition in the UI works great for one person but when extended to upto four persons we would need a RCNN which is specific to this device running in the background. However, if two or more people are standing in front of the mirror at the same time, due to the lower frames per second video evaluation, RCNN would turn out to be glitchy and would not recognize the people in the mirror.

The LDR is moderately light sensitive and over long periods of consistent, high light exposure (sunlight, lasers, bright lights, photon guns), the accuracy of the LDR would decrease and affect the display brightness adversely.

5.0 Sources Cited:

[1] Azure.microsoft.com. (2019). *Face API - Facial Recognition Software | Microsoft Azure*. [online] Available at: https://azure.microsoft.com/en-us/services/cognitive-services/face/ [Accessed 30 Aug. 2019].

[2] Amazon Web Services, Inc. (2019). *Amazon Rekognition – Video and Image - AWS*. [online] Available at: https://aws.amazon.com/rekognition/ [Accessed 30 Aug. 2019].

[3] PagerDuty Developer Documentation. (2019). *Introduction · PagerDuty Developer Documentation*. [online] Available at: https://v2.developer.pagerduty.com/docs/getting-started [Accessed 30 Aug. 2019].

[4] Developer.download.nvidia.com. (2019). *Nvidia Jetson Nano Developer Kit Datasheet*. [online]

[5] STMicroelectronics (2019). *Discovery kit for STM32 F0 microcontrollers*. STM32F0DISCOVERY. STMicroelectronics Datasheets.

[6] Capstone Connected Home. (2019). *Homepage | Capstone Connected Home*. [online] Available at: https://capstoneconnectedhome.com/?gclid=CjwKCAjwkqPrBRA3EiwAKdtwk7G4\_6JboVQtCs8plUs916RnQy1pXeI-T-Bh6pV\_BdxJLwyB9mMn0RoCRAQQAvD\_BwE [Accessed 30 Aug. 2019].

[7] Mirror.co. (2019). The Nearly Invisible Interactive Home Gym | MIRROR. [online] Available at: https://www.mirror.co/?utm\_source=google\_ads&utm\_medium=search\_cpc&utm\_campaign=brand\_mirror\_desktop&utm\_content=mirror\_co\_phrase&utm\_term=themirror [Accessed 30 Aug. 2019].

Appendix A: Functional Block Diagram

