

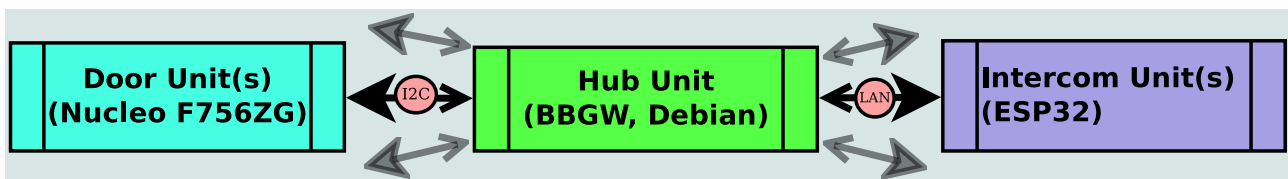
"DOORS"

*a local, centralized system
for controlling and managing
a set of automatic doors*

a prescriptive design document

1. Summary

The "Doors" system aims to elegantly fulfill a set of requirements common to most apartment buildings and many other facilities: management and monitoring of multiple physical entry-points (*Door* units), and serving of common intercom features to an arbitrary number of clients (*Intercom* units), all coordinated through a central (*Hub*) unit.

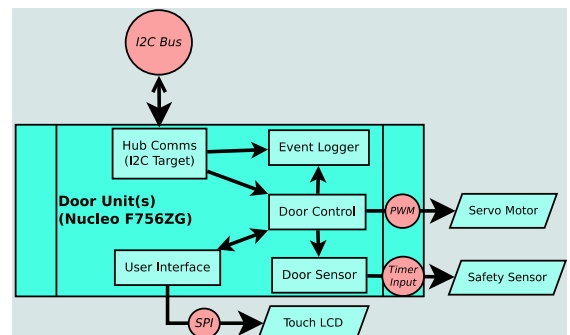


Short version: **one hub** interfaces between **multiple doors** and **multiple intercoms**.

2. The Door Unit

The *Door Unit* consists of:

- an STM32 **Nucleo-F756ZG** board;
- a **servo-motor**, controlling an automatic door;
- an additional **power supply** unit for the servo;
- a **safety sensor**, to detect blockage and avoid accidents;
- a **touch display** for passcode entry & "bell" functions;
- a small QVGA **camera**, mainly for *Intercom* functions.



Each *Door* shares an **I2C bus** with the *Hub* and all other *Door* units, and participates in the bus passively as an I2C Target.

Internally, the *Door* is a FreeRTOS based system, consisting of the following tasks:

- **User Interface**, operating the touch display and processing its inputs;
- **Hub Comms**, handling inputs from the *Hub* over the I2C bus;
- **Door Sensor**, continuously reading inputs from the safety sensor into a circular buffer;
- **Door Controller**, operating the physical door while considering input from **all above tasks**;
- **Camera**, snaps still frames in certain situations(e.g. apartment bell rung).

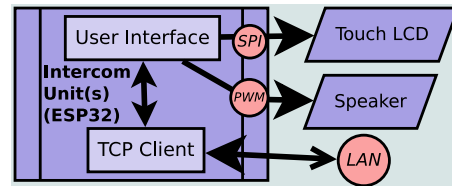
The *Door's* I2C Target module exposes four registers:

- **Event Count**, which holds the number of outgoing messages and is routinely polled by the *Hub*.
- **Event Queue**, read by the *Hub* to fetch the latest outgoing message.
- **Hub Command**, where the *Hub* writes commands for the *Door* to execute.
- **Data**, read by the *Hub* *relatively* infrequently and only after the *Door* reports that data is ready.
This would be for an unusually large packet, such as a photo from the *Door* camera.

3. The Intercom Unit

The *Intercom Unit* consists of:

- an **ESP32 WROOM-32** board;
- a **touch display** for door information & control;
- a **speaker** for the "bell" function.



Each *Intercom* unit is connected to a **local network**, and functions as a client, interacting with the singular *Hub* unit's server process to facilitate indirect communications with the *Door* layer. Internally, the *Intercom* is a simple FreeRTOS based system, with its two primary tasks being operating the user interface and communicating with the *Hub*.

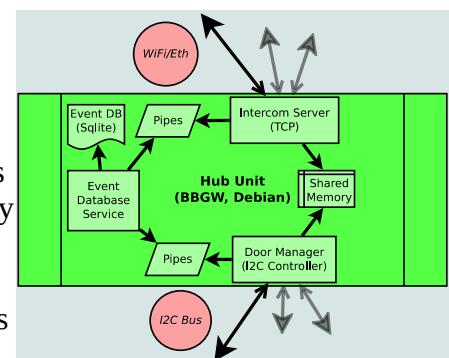
The most common use of the *Intercom* would be informing the user of a "bell" request from one of the *Door* units, while supplying additional information - a photo taken at the time of the request. The user may then use the *Intercom* unit to respond by ordering the *Door* to open.

4. The Hub Unit

The *Hub Unit* consists of a single **Beaglebone Green** board.

The board is physically connected to an **I2C bus** it shares with all *Door* units, and functions as the one and only I2C Controller of this bus. Some additional hardware, not covered here, is likely necessary for this bus to function over application-useful distances.

The *Hub* is also connected to a **local network**, where it functions as a server with all local *Intercom* units as its clients.



Internally, the *Hub* is a Debian Linux system continuously running multiple processes:

- **Door Manager**, handling I2C communication with the *Door* units;
- **Intercom Server**, handling socket communication with the *Intercom* units;
- **Event Database Service**, wrapping a local Sqlite database for access by other processes.

5. Overview

