Hola Intercom Design

# Revision History

* 09/01/2017: 1st draft
* 09/18/2017: Added buddy\_id to cloud api
* 10/02/2017: Added echo request and reply messages
* 10/16/2017: Added 3rd party components and updated messages
* 10/24/2017: Added manuf\_config.py. Updated message handler, messages and cloud\_api
* 10/25/2017: Added pin assignments. Added MAX17043 Particle Library
* 10/30/2017: Added restarted parameter to i\_am message

# Version Control

<https://github.com/rlysens/particleIntercom/>

# Third Party Components

SparkFun\_MAX17043\_Particle\_Library: <https://github.com/sparkfun/SparkFun_MAX17043_Particle_Library.git>

LCM: <https://lcm-proj.github.io/> (in source tree)

Name-gen (in source tree)

XTEA: mbedtls-2.5.1

XTEA Python: <https://pypi.python.org/pypi/xtea/0.4.0> (in source tree)

particle.py (in source tree)

* Pytz.py
* Dateutil
* Requests
* pexpect

# Shell Commands

particle call Intercom1 en|dis\_prntgrp messages/stats/default

particle serial monitor COM3/COM4

python build.py --device Intercom1/Intercom2/all [--flash]

copypython2amazon.bat

copyjson2amazon.bat

ssh2amazonec3.bat

python manuf\_config.py [-h,--help] [-c,--skip\_claim] [-f,--skip\_flash] [-i,--image\_filename <image\_filename>]

Finds and claims connected devices, sets up Wifi, flashes reference fw image and configures name, buddy name and secret key.

# Pin Assignments

Photon pin assignments

A5 = SI to codec

A4 = SO to codec

A3 = SCK to codec

A2 = xDCS to codec

A1 = xRESET to codec

A0 = DREQ to codec

DAC = xCS to codec

D0 = SDA to MAX17043 fuel gauge IC

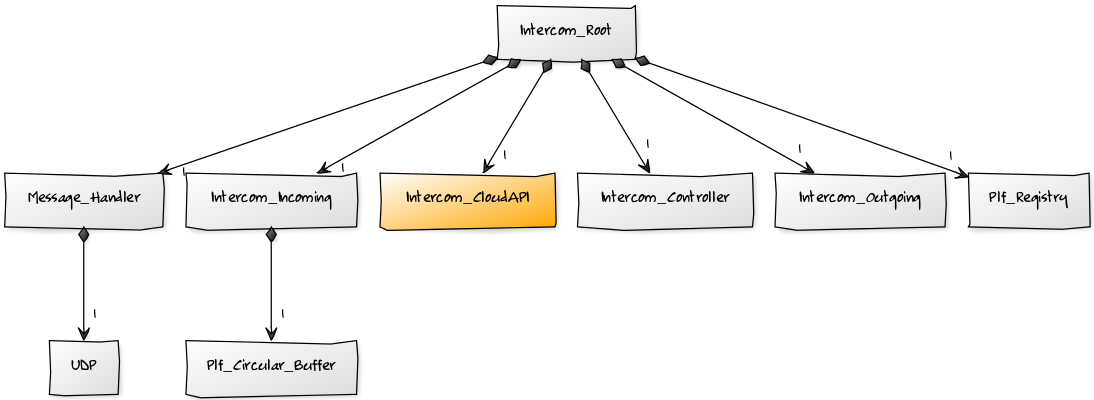
D1 = SCL to MAX17043 fuel gauge IC

D2 = Push-to-Talk Button

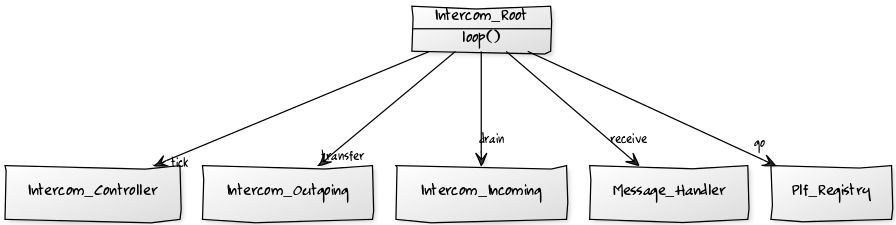
D7 = Buddy-is-Listening LED

# Firmware Design

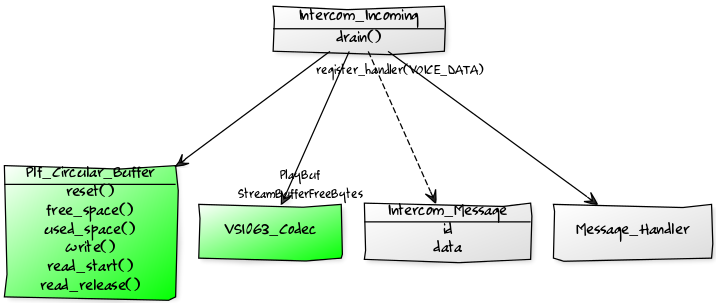
## Ownership



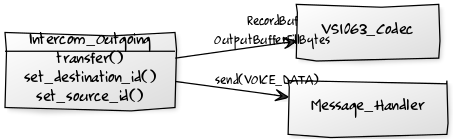
## Root



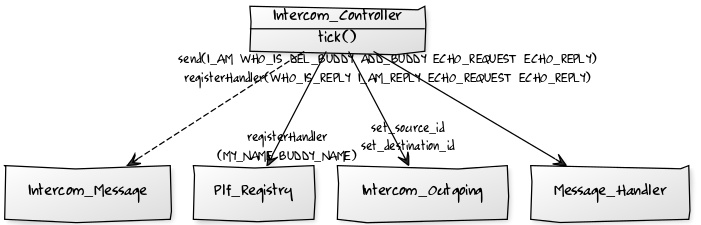
## Intercom Incoming



## Intercom Outgoing



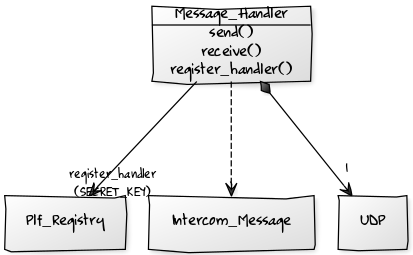
## Intercom Controller



## Codec



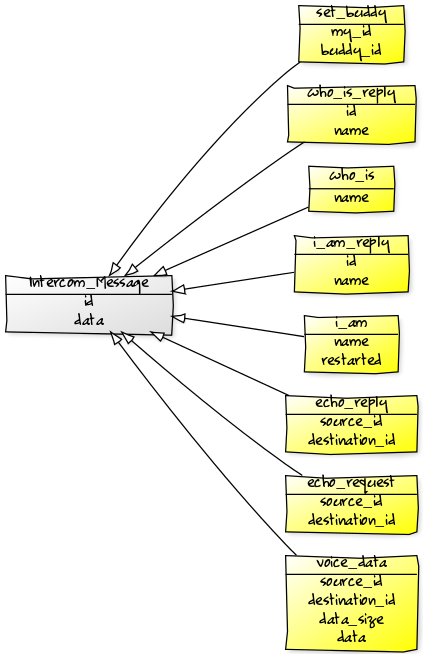
## Message Handler



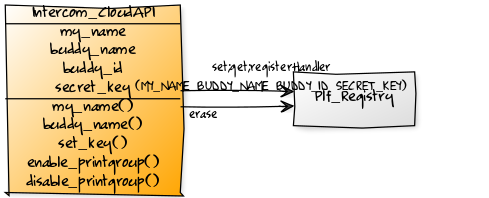
## Messages

Note that the message channel is not considered reliable. Messages can get lost and the message protocols must be robust against occasional message loss.

All messages except i\_am are encrypted using XTEA block cipher. The secret key is configured into the device at manufacturing time.



## Intercom Cloud API



# Registry keys

REG\_KEY\_MY\_NAME persistent

REG\_KEY\_BUDDY\_NAME persistent

REG\_KEY\_BUDDY\_ID volatile

REG\_KEY\_SECRET\_KEY persistent

# Server Side

**MSG\_TABLE**

msg\_id -> msg\_handler\_fun

**Intercom\_name\_to\_id\_table**

Intercom\_name -> intercom\_id

**Intercom\_id\_to\_intercom\_table**

Intercom\_id -> intercom

**Message\_Handler**

**Send()**

**Receive()**

socket

**Intercom**

**sendTo()**

**addBuddy()**

**delBuddy()**

**getEncoderCryptoCodec()**

**getDecoderCryptoCodec**

Msg\_voice\_data\_handler

Msg\_i\_am\_  
handler

Msg\_who\_is\_  
handler

Add\_buddy\_  
handler

del\_buddy\_  
handler

1..1

n..1

1..1

Msg\_echo\_request\_data\_handler

Msg\_echo\_reply\_data\_handler

# Manuf\_config.py

**Credentials.json**

**Name\_key.json**

**Manuf\_config.py**

Particle.py

Name\_key\_gen.py

Namegen.py