## Inhaler Cap Design

* The current design uses Bluetooth 2.1
* New design will use Bluetooth 4.0 module to extend battery life. It is not supported in old phones.  
  The link to Bluetooth 4.0 is : <http://www.bluetooth.com/Pages/Bluetooth-Smart-Devices-List.aspx>

## When device is turned on:

1. Start listening to button presses.

## When discovery mode button is pressed:

1. Enter discovery mode
2. Possibly flash LED?

## When Bluetooth is paired:

1. Disable broadcasting to other devices. (only stay paired to one device)

## When GetData command is received from phone:

1. Send all timestamps in memory to phone over Bluetooth.
2. Send the number of timestamps sent to the phone over Bluetooth
3. Wait for phone to acknowledge that it received the all the timestamps.

## When DeleteData command is received from phone:

1. Delete all timestamps in memory
2. Send acknowledgement to phone over Bluetooth.

## When GetStatus command is received from phone:

1. Read battery voltage level.
2. Convert to a percentage and send to phone.

## When inhaler is used:

1. Get current time
2. Store data in memory

## Battery Life:

Using a CR2032 coin cell battery (non-rechargeable) capacity is 210mAh.

Power mode 1 = 270uA

mode 2 = 1uA

mode 3 = .5uA

Ideally if the device can stay in power mode 1 most of the time it should consume <300uA.

Ideally if the device can stay in power mode 2 or 3 most of the time it should consume <10uA.

## Timer:

I am not aware of any way to get the actual time from the Bluetooth module without asking the phone for the time. The issue with this is that asking the phone for the time obviously does not work when the device is not connected to the phone. The way I plan to get around this is have a counter. When the inhaler is used store the use as a 0 in memory. Then every X interval add 1. That way when the phone receives the data it can trace back how long ago the data was used.

To keep track of the intervals there are two options.

Option 1 is to attempt to keep track of time using internal oscillator.

Pros

* No extra parts, reduced size.

Cons

* Not as accurate as external oscillator.
* Could prevent the device from entering certain power modes.

Option 2 is to attempt to keep track of time using external oscillator.

Pros

* Better accuracy than internal oscillator.
* More likely to allow lower power modes than internal oscillator.

Cons

* Extra part, increase in size.

## Programming setup:

In order to program the BLE113 chip 7 pins need to be connected to the programmer.

BLE113 - CC Debugger

GND – GND

AVDD – Target Voltage Sense

AVDD – 3.3V

DVDD – Target Voltage Sense

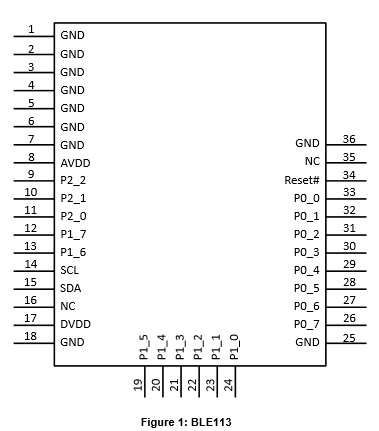
DVDD – 3.3V

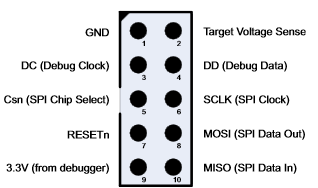
P2-2 – Debug Clock

P2\_1 – Debug Data

P0\_0 – RESETn

Reset – RESETn





CC Debugger

BLE113

# Prototype sent to Sada

## Wiring

There should be one black wire and one red wire attached to the breadboard. These are the wires that are supposed to be attached to a power supply. The end of each wire closest to the center, or by the BLE113 chip, is the end that stays connected to the breadboard. The Other end of the red and black wire are attached to opposite ends of the breadboard to keep them from falling out. Those ends can be removed from the breadboard and attached to a power supply. The red wire connects to 3V and the black wire connects to ground.

## Function

The BLE113 module runs a GATT service with four characteristics.

|  |  |  |  |
| --- | --- | --- | --- |
| Name | R/W | UUID | Function |
| Recent Use | R | 00431c4a-a7a4-428b-a96d-d92d43c8c7cb | Time of most recent button press |
| Command | W | 00431c4a-a7a4-428b-a96d-d92d43c8c7cc | Remote commands |
| Count | R | 00431c4a-a7a4-428b-a96d-d92d43c8c7cd | Number of saved button presses |
| Time | R | 00431c4a-a7a4-428b-a96d-d92d43c8c7ce | Number of seconds |

### On button press:

1. BLE113 wakes up
2. Count is increased by 1
3. Recent Use is set to the current Time
4. Recent Use is saved to memory
5. BLE113 goes to sleep

### Every 1 second:

1. BLE113 wakes up
2. Time is increased by 1
3. Check for long button press
4. BLE113 goes to sleep

### On long button press (≈5 seconds):

1. Advertise for 30 seconds

### When 0x0000 is written to Command:

1. BLE113 wakes up
2. Recent Use is erased from memory
3. \*Count is decreased by 1
4. Next Recent Use is loaded from memory
5. BLE113 goes to sleep

\*If there are no more saved uses then everything is reset, similar to sending 0x1111 command

### When 0x1111 is written to Command:

1. BLE113 wakes up
2. Recent Use is set to 0
3. Count is set to 0
4. Time is set to 0
5. BLE113 goes to sleep