WiPy Tools Documentation

Release 0.0.165

Dwight Hubbard

May 30, 2018

Contents

1 Table of Contents **1**

1.1 Quickstart 1
1.2 Start the service 1
1.3 Configure the cloudclient on the micropython boards to talk to the service 1
1.4 List the boards 2
1.5 Run a command on some boards 2
1.6 Upload a file to some boards 2
1.7 Installation Instructions 3
1.8 Design 3
1.9 Command Line Interface 4

2 Indices and tables **9**

CHAPTER 1

Table of Contents

1.1 Quickstart

1.1.1 Requirements

The micropython-cloudmanager does not currently have the security and validation functionality implemented. As a result it should only be used on isolated secure networks.

1.1.2 Install

The micropython-cloudmanager should run on any Posix compliant operating system that is supported by redis.

However some functionality is only available when running on a fairly current Linux operating system.

The micropython-cloudmanager is easiest to install from pypi with pip:

```
pip install micropython-cloudmanager
```

1.2 Start the service

Run the *mbm server-start* command to start the cloudmanager service on the current host:

```
$ mbm server-start
```

1.3 Configure the cloudclient on the micropython boards to talk to the service

Install and configure the micropython-redis-cloudclient on the micropython boards to be managed.

For esp8266 boards such as nodemcu, wemos-d1 boards the easiest method Using the directions at https://github.com/dwighthubbard/micropython-redis-cloudclient/blob/master/README.md

1.4 List the boards

The *mbm board-list* command will list the boards that have registered with the cloudmanager service:

```
$ mbm board-list
Name       Platform                                            State
esp8266-1  esp8266                                             idle
esp8266-2  esp8266                                             idle
esp8266-3  esp8266                                             idle
wipy2-1    WiPy                                                idle
$
```

1.5 Run a command on some boards

The *mbm board-execute* command will send the commands from standard input to one or more boards.

Run the *mbm board-execute [boardname]* command, then type the code to execute and hit **CTRL-D** and the code will be sent to he board(s), executed and the results displayed:

```
$ mbm board-execute esp8266-2
import os
print(os.uname())
## Executing on 'esp8266-2' ################################
(sysname='esp8266', nodename='esp8266', release='1.5.4(baaeaebb)', version='v1.8.5-
↪100-g10bde69-dirty on 2016-11-01', machine='ESP module with ESP8266')

$
```

1.6 Upload a file to some boards

The *mbm board-upload* command will upload a file to one or more boards.

So for example to copy the file “hello_world.py” to the lib (module) directory on 2 boards works like this:

```
$ mbm board-upload esp8266-[1-2],wipy2-1 hello_world.py lib/hello_world.py
$ mbm board-execute esp8266-[1-2],wipy2-1
import hello_world
hello_world.hello_world()
## Executing on 'esp8266-1' ################################################
Hello World!

## Executing on 'esp8266-2' ################################################
Hello World!

## Executing on 'wipy2-1' ###################################################
Hello World!

$
```

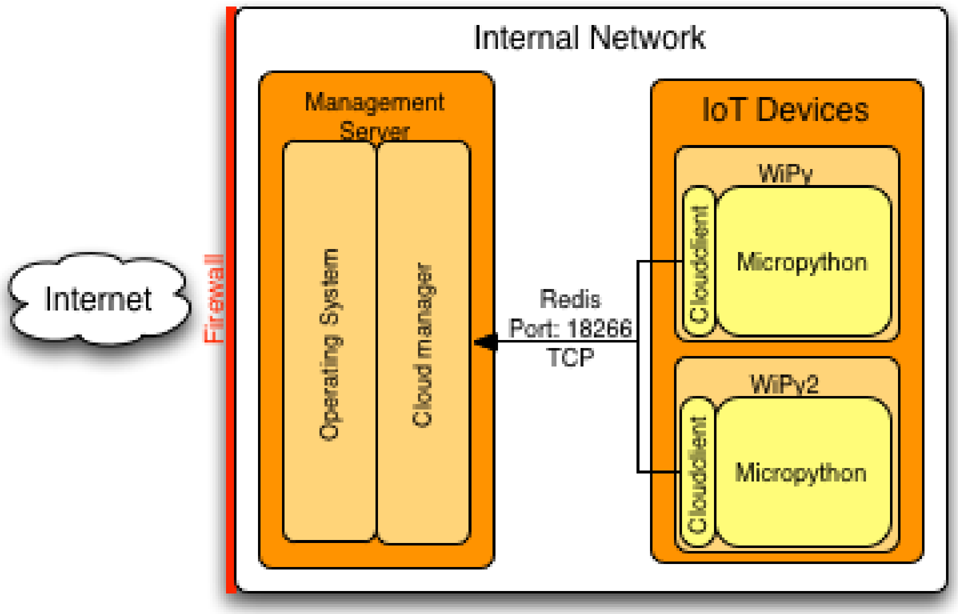
1.7 Installation Instructions

```
pip install micropython-cloudmanager
```

1.8 Design

Cloudmanager uses a central management hub that relays requests from the management api to the clients.

All clients of the cloudmanager connect in to the central cloudmanager server.



1.8.1 Philosophy

Here are the overall design concepts that are the focus of the cloudmanger. Which helps to explain it's purpose and some guidance on where it is going in the future.

All Python Infrastructure

If a programmer is writing code to operate on boards running Micropython. It is desirable to be able to also write the service management code in the same language.

Simple to set up a basic configuration

A goal of this service is to be simple to set up a basic functional configuration.

To meet this goal the command line interface tool has commands to start/stop and check the status of the service with reasonable defaults without any other configuration or setup.

In addition we provide a seperate flash utility that will flash popular esp8266 boards with micropython and configure them as cloudmanager clients with a single command.

Architecture to minimize security attack surface

Security is important and when new security attacks occur frequently new code has to be added to deal with the issues. This can be difficult to do when the code to handle the attack has to be implemented on an IoT board with little free memory.

To address this issue, the cloudmanager is designed to have a single netowrk entrypoint that accepts incomming network connections.

This provides a single network point to secure, which provides a number of benefits.

- It minimizing the attack surface.
- Lessens requirements to update IoT board software for security issues
- It moves most of the processing for authentication, and input validation to the cloudmanager service node which generally will have significantly more resources to handle secufity issues properly.

Do resource intensive operations on the server not the IoT devices

The client should provide the minimum functionality needed to implement the functionality. In addition functionality should be added with a focus on performing resource intensive operation on the managment nodes and not on the IoT boards that have minimal resources.

For example, the **mbm board-install** command installs micropython packages on boards. The implementation of this functionality performs the resource intensive download, unpack, and dependency handling on the cloudmanager server. The only function performed on the board is the upload to the appropriate location in the boards filesystem.

1.9 Command Line Interface

The micropython cloud manager command line interface is managaed using the `mbm` command line utility on the main service host.

The `mbm` utility supports managing the cloud service as well as performing operations on boards running micropython that are connecting to the cloud management server.

1.9.1 Cloudmanager Service Management Comands

Before any commands can be run the service needs to be available.

service-start

This command will start the cloudmanager service as a daemon process running as the user that starts it.

```
$ mbm server-start
Cloudmanager service is listening on: 192.168.1.127:18266
$
```

service-stop

This command stops the running cloudmanager service.

```
$ mbm server-stop
Service is shutdown
$
```

service-status

This command shows the status of the cloudmanager service.

```
$ mbm server-status
Running
$
```

1.9.2 Cloudmanager Board (client) commands

The Cloudmanager board commands are used to interact with boards running the cloudmanager client.

All commands that talk to boards support hostlists range operators when specifying the boards to operate on. This makes it simple to perform the same operation on multiple boards.

For example:

```
esp8266-0[08-11],wipy2-[1,3,7]
```

Will operate on this list of boards:

```
esp8266-008
esp8266-009
esp8266-010
esp8266-011
wipy2-1
wipy2-3
wipy2-7
```

board-scan

The board-scan command will scan for unconfigured micropython boards running the cloudclient.

Note - This command currently only works on Linux

board-list

The board-list command shows all boards that are registered with the cloudmanager service.

```
$ mbm board-list
Name        Platform                                           State
esp8266-1   esp8266                                            idle
esp8266-2   esp8266                                            idle
esp8266-3   esp8266                                            idle
wipy2-1     WiPy                                               idle
$
```

board-rename

The board-rename command will change the name of a managed board

usage: mbm board-rename [-h] board new_name

positional arguments: board Board to rename new_name New board name

$ mbm board-rename esp8266-5 esp8266-2 $

board-execute

The board-execute command will send the command from the stdin stream to all the boards specified and return the output

usage: mbm board-execute [-h] board

positional arguments: board Range of board(s) to execute the code on

```
$ mbm board-execute --debug esp8266-[1-3]
print('hello')
## Executing on 'esp8266-1' ####################################################
hello

## Executing on 'esp8266-2' ####################################################
hello

## Executing on 'esp8266-3' ####################################################
hello

$
```

board-upload

The board-upload command will upload a file to all of the specified boards.

usage: mbm board-upload [-h] board filename dest

positional arguments: board Range of board(s) to upload to filename File to upload dest Destination filename (optional)

```
$ mbm board-upload esp8266-[1-3] example_file example_file
Copying file to esp8266-1:example_file
Copying file to esp8266-2:example_file
Copying file to esp8266-3:example_file
$ mbm board esp8266-[1-3] ls
## 'ls' on 'esp8266-1' ##########################################################
boot.py
etc
main.py
example_file
## 'ls' on 'esp8266-3' ##########################################################
boot.py
etc
main.py
example_file
## 'ls' on 'esp8266-2' ##########################################################
boot.py
etc
```

(continues on next page)

(continued from previous page)

```
main.py
example_file
$
```

board-install

The board-install package will intall a package on the board(s) specified.

Note: This command performs the download and unpack of the package files on the cloudmanager server so does not require upip or it's dependencies be installed on the boards being operated on.

```
$ mbm board-install esp8266-[1-3] micropython-logging
Installing package 'micropython-logging'
Copying file to esp8266-1:lib/logging.py
Copying file to esp8266-3:lib/logging.py
Copying file to esp8266-2:lib/logging.py
$ mbm board-execute esp8266-[1-3]
import logging
logging.info('Example log message')
## Executing on 'esp8266-1' #####################################################
INFO:None:Example log message

## Executing on 'esp8266-3' #####################################################
INFO:None:Example log message

## Executing on 'esp8266-2' #####################################################
INFO:None:Example log message

$
```

CHAPTER 2

Indices and tables

- genindex
- modindex
- search