

## Micropython Based On AT32 RTT

## 前言

本文档主要讲述如何在基于 AT32 RT-Thread 的源码包中开启并下载 Micropython 开源包，并简单描述和展示了 Micropython 在 AT32 系列 MCU 上正常运行起来后的现象及 python 语句的执行结果。

支持型号列表：

|      |          |
|------|----------|
| 支持型号 | AT32F 系列 |
|------|----------|

## 目录

|          |                                               |           |
|----------|-----------------------------------------------|-----------|
| <b>1</b> | <b>AT32 RTThread Micropython 快速使用方法 .....</b> | <b>5</b>  |
| 1.1      | 硬件资源 .....                                    | 5         |
| 1.2      | 软件资源 .....                                    | 5         |
| <b>2</b> | <b>menuconfig 配置开启 Micropython .....</b>      | <b>6</b>  |
| <b>3</b> | <b>在线 download Micropython 包 .....</b>        | <b>8</b>  |
| <b>4</b> | <b>生成工程文件或编译。 .....</b>                       | <b>9</b>  |
| <b>5</b> | <b>版本历史 .....</b>                             | <b>10</b> |

## 表目录

|                   |    |
|-------------------|----|
| 表 1. 文档版本历史 ..... | 10 |
|-------------------|----|

## 图目录

|                                |   |
|--------------------------------|---|
| 图 1. AT-START-F403A V1.0 ..... | 5 |
| 图 2 online packages .....      | 6 |
| 图 3 language packages .....    | 6 |
| 图 4 Micropython 配置项 .....      | 7 |
| 图 5 pkgs 源更新 .....             | 8 |
| 图 6 pkgs 下载.....               | 8 |
| 图 7 Micropython 源码包 .....      | 8 |

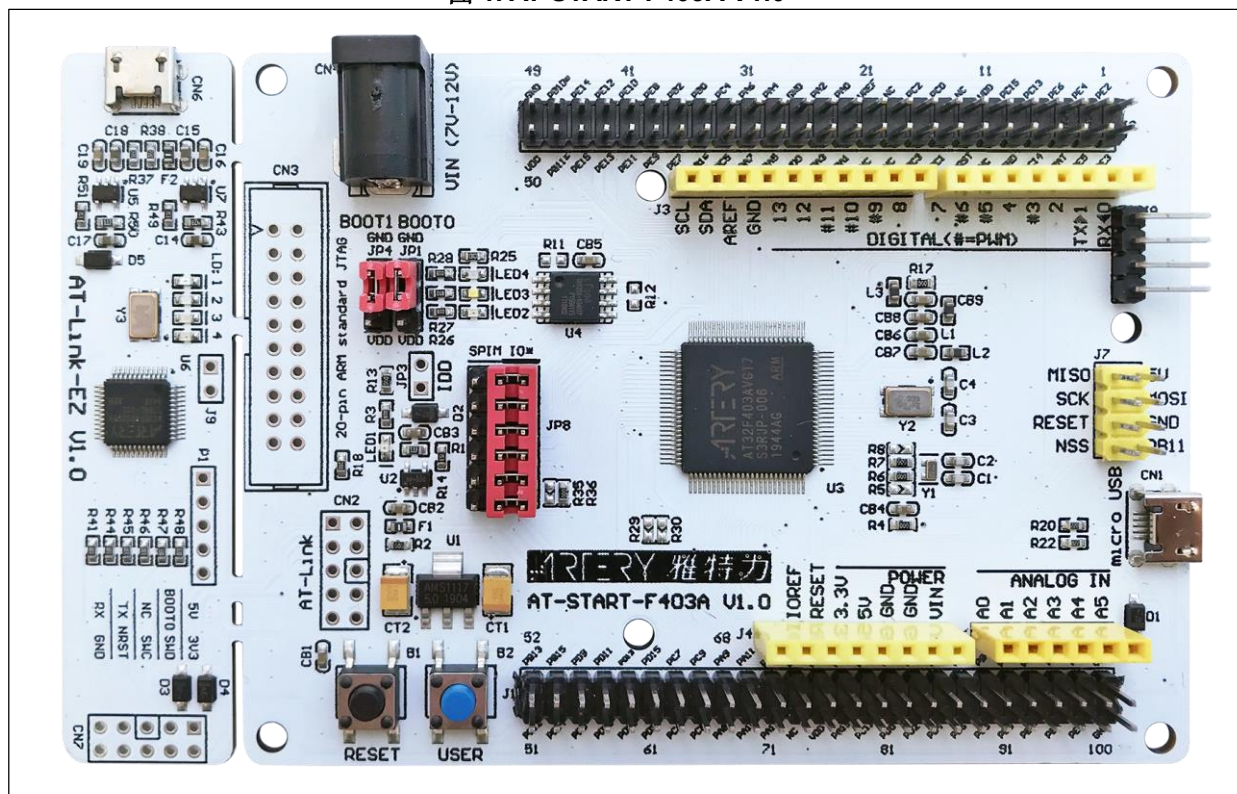
## 1 AT32 RTThread Micropython 快速使用方法

### 1.1 硬件资源

以 AT-START-F403A 示例，其资源如下：

- 1) 指示灯 LED2/LED3/LED4
- 2) USART1(PA9/PA10)
- 3) AT-START-F403A V1.0 实验板

图 1. AT-START-F403A V1.0



### 1.2 软件资源

rt-thread 源码请到 <https://github.com/rt-thread> 进行下载。

AT32 支持 RTThread 的 BSP 包在 rt-thread 的存放目录结构为 xxx/rt-thread/bsp/at32。

**Note:** *rt-thread* 源码包的使用方法请采用 *rt-thread* 官方提供的 ENV 工具及用户手册进行配置和使用。

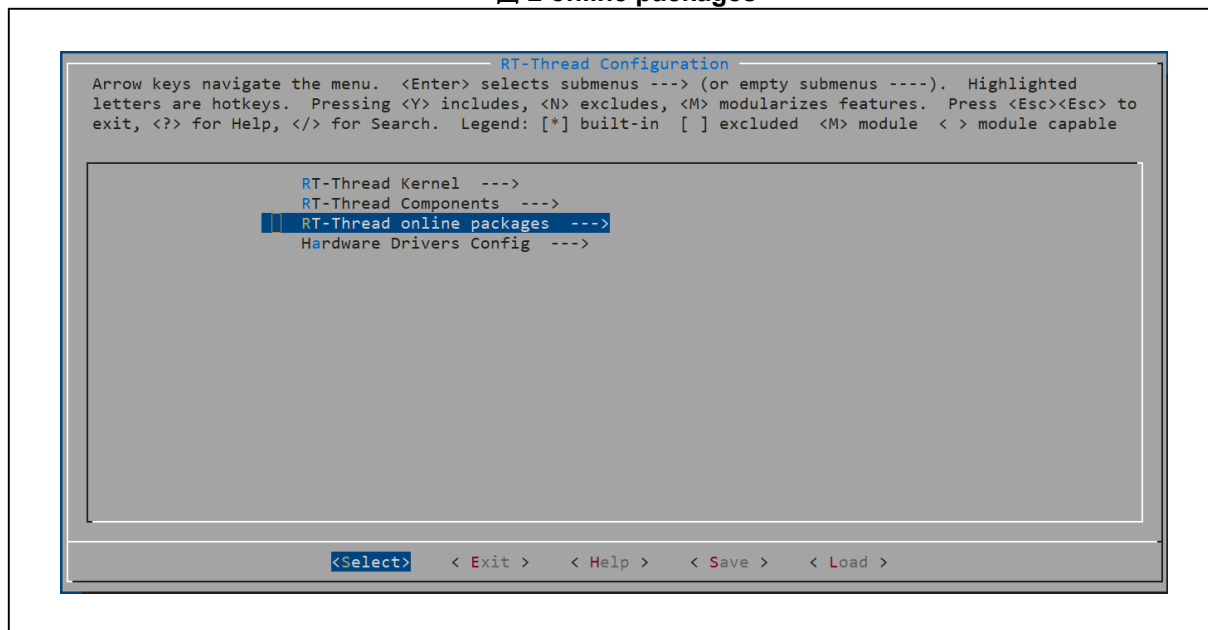
## 2 menuconfig 配置开启 Micropython

打开 ENV 工具，以下以 at32f403a-start 为例，切换目录到 at32f403a-start 包目录下，运行 menuconfig。按 menuconfig 操作方式逐级进入并打开 Micropython 配置项。流程如下：RT-Thread online packages ----> language packages----> [\*]MicroPython: A lean and efficient Python implementation for ...

图示及操作如下：

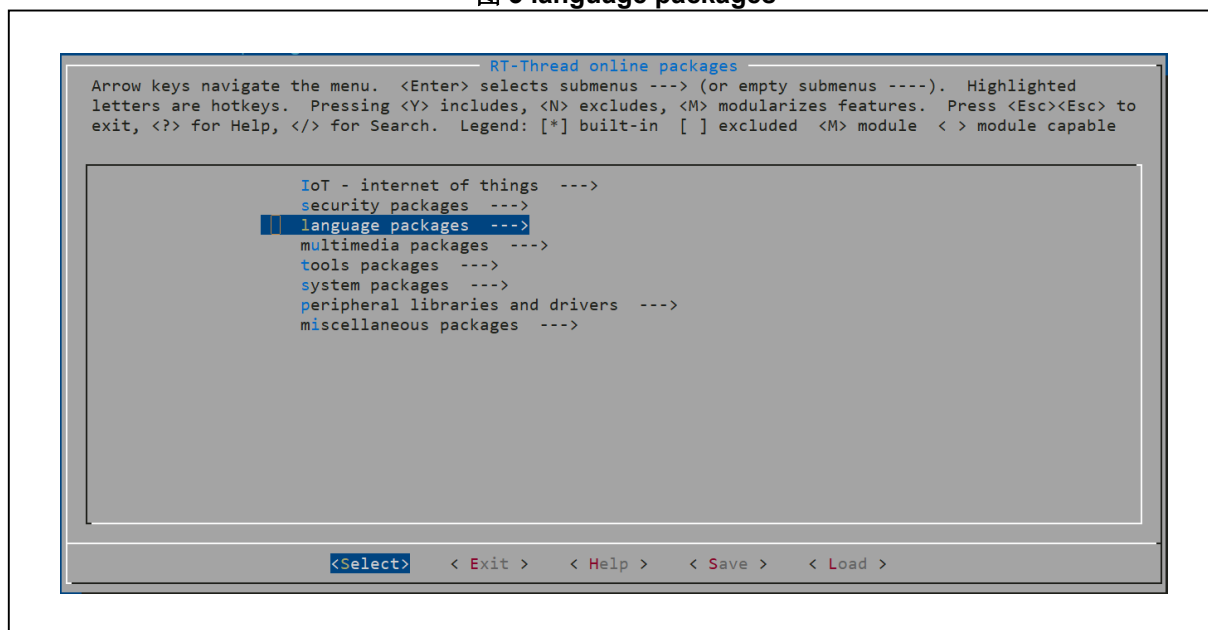
- 1、打开 menuconfig 后可看到如下图例，按键盘方向下键移动光标到 RT-Thread online packages 并点击回车键

图 2 online packages



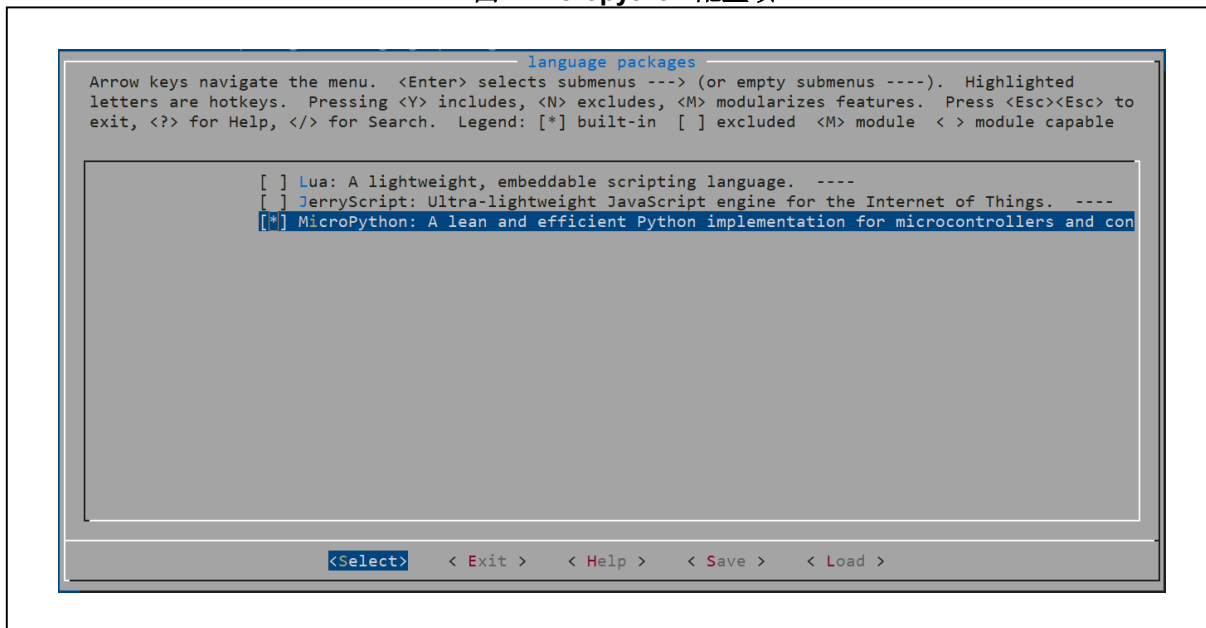
- 2、显示如下界面时，使用键盘方向下键移动光标到 language packages 并点击回车键

图 3 language packages



- 3、显示如下界面时，将光标移动到 MicroPython: A lean and efficient Python ...并点击键盘空格键， [\*]即表示配置已选中。逐级退出 menuconfig 并保存配置。

图 4 Micropython 配置项



注：为保证 python 能正常运行，最好确保将启动 python 的线程的堆栈大小配置为 $\geq 8192$ 。如在 shell 中来使用 python，则配置 RT-Thread Components ---> Command shell ---> (4096)The stack size for finish thread。选中此处回车后的对话框中将 4096 修改为 $\geq 8192$

## 3 在线 download Micropython 包

注：需保证外网连接正常

1、使用命令 `pkgs --upgrade`，运行成功后显示如下：

图 5 pkgs 源更新

```
sheltonyu@ATK-NB001 C:\Shelton\RTThreadResp\rt-thread\bsp\at32\at32f403a-start
$ pkgs --upgrade
Begin to upgrade env packages.
From https://gitee.com/RT-Thread-Mirror/packages
* branch          HEAD          -> FETCH_HEAD
=====> Env packages upgrade done

Begin to upgrade env scripts.
From https://gitee.com/RT-Thread-Mirror/env
* branch          HEAD          -> FETCH_HEAD
=====> Env scripts upgrade done
```

2、再使用命令 `pkgs --update`，运行成功后显示如下：

图 6 pkgs 下载

```
sheltonyu@ATK-NB001 C:\Shelton\RTThreadResp\rt-thread\bsp\at32\at32f403a-start
$ pkgs --update
=====> MICROPYTHON v1.10.4 is downloaded successfully.

Operation completed successfully.
```

3、当 Micropython 包下载完成后可在 `xxx/rt-thread/bsp/at32/at32f403a-start` 目录下看到新增了一个 `packages` 文件夹，点击打开即可看到新下载的 `micropython-v1.10.4` 源码包就存放在此。内容如下：

图 7 Micropython 源码包

|                                                                                                                      |                 |             |
|----------------------------------------------------------------------------------------------------------------------|-----------------|-------------|
|  <code>micropython-v1.10.4</code> | 2020/3/30 15:59 | 文件夹         |
|  <code>packages.dbsqlite</code>   | 2020/3/30 15:59 | DBSQLITE 文件 |
|  <code>pkgs.json</code>           | 2020/3/30 15:59 | JSON 文件     |
|  <code>pkgs_error.json</code>     | 2020/3/30 15:59 | JSON 文件     |
|  <code>SConscript</code>          | 2020/3/30 15:59 | 文件          |

## 4 生成工程文件或编译

在 ENV 控制台输入如下命令可生成 IDE 开发工程和编译。

|                    |                    |
|--------------------|--------------------|
| scons -target=mdk5 | 生成 keil_v5 工程文件    |
| scons -target=mdk4 | 生成 keil_v4 工程文件    |
| scons -target=iar  | 生成 iar 工程文件        |
| scons              | 使用 gcc 交叉编译工具链进行编译 |

编译完成后下载运行。PC 串口接上 USART1，在串口控制台输入 help 命令（或串口工具发送 help 字符串）后可看到有关 python 的命令及介绍。操作及内容如下：

```

\\ /
- RT -      Thread Operating System
/| \       4.0.3 build Mar 30 2020
2006 - 2020 Copyright by rt-thread team
msh />help
RT-Thread shell commands:
python      - MicroPython: `python [file.py]` execute python script
list_fd     - list file descriptor
date        - get date and time or set [year month day hour min sec]
version     - show RT-Thread version information
list_thread - list thread
list_sem    - list semaphore in system
list_event  - list event in system
list_mutex  - list mutex in system
list_mailbox - list mail box in system
list_msgqueue - list message queue in system
list_memheap - list memory heap in system
list_mempool - list memory pool in system
list_timer  - list timer in system
list_device - list device in system
...

```

当输入 python 命令即可看到 MicroPython 运行及显示了当前 Micropython 的版本信息等，再在 Python 控制台输入 python 命令 print("hello world")，可看到 python 能正常打印，操作及显示如下：

```

python

MicroPython v1.10.3.99 on 2019-09-24; Universal python platform with RT-Thread
Type "help()" for more information.
>>> print("hello world")
hello world
>>>

```

如需深入测试与使用可编写 python 脚本来运行，举例如现有 test.py 可运行，可在 console 输入命令，执行结果以脚本内容为准，操作如下：

```

\\ /
- RT -      Thread Operating System
/| \       4.0.3 build Mar 30 2020
2006 - 2020 Copyright by rt-thread team
msh />python test.py
...

```

## 5 版本历史

表 1. 文档版本历史

| 日期         | 版本    | 变更   |
|------------|-------|------|
| 2022.04.08 | 2.0.0 | 最初版本 |

**重要通知 - 请仔细阅读**

买方自行负责对本文所述雅特力产品和服务的选择和使用，雅特力概不承担与选择或使用本文所述雅特力产品和服务相关的任何责任。

无论之前是否有过任何形式的表示，本文档不以任何方式对任何知识产权进行任何明示或默示的授权或许可。如果本文档任何部分涉及任何第三方产品或服务，不应被视为雅特力授权使用此类第三方产品或服务，或许可其中的任何知识产权，或者被视为涉及以任何方式使用任何此类第三方产品或服务或其中任何知识产权的保证。

除非在雅特力的销售条款中另有说明，否则，雅特力对雅特力产品的使用和/或销售不做任何明示或默示的保证，包括但不限于有关适销性、适合特定用途（及其依据任何司法管辖区的法律的对应情况），或侵犯任何专利、版权或其他知识产权的默示保证。

雅特力产品并非设计或专门用于下列用途的产品：（A）对安全性有特别要求的应用，例如：生命支持、主动植入设备或对产品功能安全有要求的系统；（B）航空应用；（C）航天应用或航天环境；（D）武器，且/或（E）其他可能导致人身伤害、死亡及财产损害的应用。如果采购商擅自将其用于前述应用，即使采购商向雅特力发出了书面通知，风险及法律责任仍将由采购商单独承担，且采购商应独立负责在前述应用中满足所有法律和法规要求。

经销的雅特力产品如有不同于本文档中提出的声明和/或技术特点的规定，将立即导致雅特力针对本文所述雅特力产品或服务授予的任何保证失效，并且不应以任何形式造成或扩大雅特力的任何责任。

© 2022 雅特力科技 保留所有权利