# MCUSH扩展命令使用说明

说明：此文档适用于MCUSH平台提供的扩展、非通用型命令。

作者：彭树林

更新日期：2019-9-25

## 1、硬件控制类

### sgpio

别名：无

功能：控制DMA驱动的GPIO

语法： sgpio [-p <port\_index>] [-o <output\_mode>] [-i <input\_mode>] [--input\_len=<input\_buf\_len>] [-f <frequency>] [-l] [-r] [-s] [--info]

options:

-p/--port index from 0

-o/--output set output mode mask

-i/--input set input mode mask

--input\_len length of input buffer

-f/--frequency 1~1000000hz

-l/--loop loop mode

-r/--start run

-s/--stop stop

--info print info

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>sgpio -p2 -o 0xFF -f 10 -l  >1 2 4 8 16 32 64 128  >  =>sgpio --start  => | 用sgpio控制PC0~7实现流水灯输出 |
| =>sgpio --info  run port:0 output:0x000F input:0x0000 loop:1 frequency:10.0  buf\_out:0x20002B48 buf\_in:0x00000000 length:8  => | 显示sgpio状态 |
| =>sgpio --stop  => | 停止 |
| =>sgpio -p0 -i0xFF -f10 --input\_len=100  =>sgpio --start  =>sgpio --info  run port:0 output:0x0000 input:0x00FF loop:0 frequency:10.0  buf\_out:0x00000000 buf\_in:0x20002B60 length:100  =>sgpio --info  stop port:0 output:0x0000 input:0x00FF loop:0 frequency:10.0  buf\_out:0x00000000 buf\_in:0x20002B60 length:100  =>x -b 0x20002B60 -w2 -l200  20002B60: B5FF B5FF ADFF B5FF B5FF B5FF B5FF B5FF  20002B70: B5FF B5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002B80: B5FF ADFF A5FF B5FF B5FF B5FF B5FF B5FF  20002B90: B5FF B5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002BA0: B5FF A5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002BB0: B5FF B5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002BC0: B5FF ADFF B5FF B5FF B5FF B5FF B5FF B5FF  20002BD0: B5FF B5FF B5FF B5FF B5FF B5FF B5FF ADFF  20002BE0: ADFF B5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002BF0: B5FF B5FF B5FF B5FF B5FF ADFF B5FF B5FF  20002C00: B5FF B5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002C10: B5FF B5FF B5FF B5FF B5FF B5FF B5FF B5FF  20002C20: B5FF B5FF B5FF B5FF  => | 用sgpio采集PA0~7的8路IO口，采样频率10Hz，长度100（10秒）  测量结束  打印测量结果（100点） |

注：需BSP支持。

### pwm

别名：无

功能：控制硬件看门狗

语法： pwm [-i <pwm\_index>] [-v <value>] [-f <frequency>] [-r <range>] [-n] [-I] [-D]

options:

-i/--index index from 0

-v/--value value param

-f/--frequency 1~100000(default 1000)hz

-r/--range default 100

-n/--number query

-I/--init init

-D/--deinit deinit

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>pwm -I  =>pwm -i 0 -v 25  =>pwm -i 1 -v 50  =>pwm -i 1 -v 75 | 控制PWM输出频率1kHz，占空比分别为25%、50%、75%的3通道信号 |
| =>pwm -D  => | 停止输出，释放资源。 |

注：需BSP支持。

### adc

别名：无

功能：控制ADC

语法： adc [-l <loop\_delay\_ms>] [-i <channel\_index>] [-I] [-D]

options:

-l/--loop default 1000ms

-i/--index select channel

-I/--init init

-D/--deinit deinit

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>adc -I  =>adc  3.30,2.68,2.21,1.86,1.59,1.39,1.25,1.13  => | 打印所有通道ADC测量值 |
| =>adc -l100  3.30,2.68,2.21,1.88,1.60,1.42,1.29,1.19  3.30,2.68,2.22,1.84,1.60,1.41,1.29,1.20  3.30,2.67,2.20,1.86,1.60,1.40,1.27,1.15  3.30,2.68,2.21,1.87,1.59,1.39,1.26,1.13  3.30,2.68,2.22,1.88,1.59,1.40,1.27,1.15  ...  3.30,2.68,2.21,1.88,1.60,1.41,1.29,1.18  3.30,2.68,2.22,1.88,1.61,1.43,1.33,1.24  => | 同上，循环打印，Ctrl-C中止 |
| =>adc -D  => | 停止测量，释放资源 |

注：需BSP支持。

### ws2812

别名：W

功能：控制ws2812全彩色LED芯片

语法： ws2812 [-l <length>] [-D] [-w] [-g] [-o <offset>] <data> [-p <pin>] [-I]

options:

-l/--length total pixels

-D/--deinit deinit

-w/--write write

-g/--grb GRB instead of RGB

-o/--offset data offset

-p/--pin default 0.0

-I/--init init

data data to be written

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>W -l 3 -I  =>W -o 0 -w 0xFF0000 0x00FF00 0x0000FF  => | 初始化一个长度3的LED灯带  分别设置成红/绿/蓝三色 |
| =>W -D  => | 释放资源 |

注：建议配合Python库实现自动化控制

### ds1w

别名：D

功能：控制Dallas-1Wire总线

语法： ds1w [-r] [-z] [-w <(null)>] [-0] [-1] [-R] [-p <pin>] [-I]

options:

-r/--read read

-z/--r0 read bit 0

-w/--write write

-0/--w0 write bit 0

-1/--w1 write bit 1

-R/--reset reset

-p/--pin default 0.0

-I/--init init

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>D -I  => | 初始化 |
| =>D -R  => | 总线复位 |

注：建议配合Python库实现自动化控制（总线设备搜索，控制单个设备）。

### dht11

别名：无

功能：控制奥松生产的温湿度测量模块

语法： dht11 [-p <pin>]

options:

-p/--pin default 0.0

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>dht11  40.0  20.0  => | 测量返回：  第一行：湿度  第二行：温度 |
| =>dht11  not present  !> | 模块异常时 |

注：建议配合Python库实现自动化控制

### lan8720

别名：无

功能：控制以太网PHY芯片lan8720

语法： lan8720 [-c <command>] [-n <name>] [-v <value>]

options:

-c/--command info|reset|read|write|down|loop

-n/--name name param

-v/--value value param

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>lan8720  BCR: 1100  BSR: 782D  IDN1: 0007  IDN2: C0F1  NADV: 01E1  NLPA: 4DE1  NEXP: 0003  MODE: 0002  SMOD: 60E0  SEC: 0000  SCSI: 000A  INTS: 00C8  INTM: 0000  SCS: 1058  => | 打印内部寄存器：  BCR: Basic Control Register  BSR: Basic Status Register  IDN1: PHY Identifier 1  IDN2: PHY Identifier 2  NADV: Auto-Negotiation Advertisement Register  NLPA: Auto-Negotiation Link Partner Ability Register  NEXP: Auto-Negotiation Expansion Register  MODE: Mode Control/Status Register  SMOD: Special Modes  SEC: Symbol Error Counter Register  SCSI: Control/Status Indication Register  INTS: Interrupt Source Register  INTM: Interrupt Mask Register  SCS: PHY Special Control/Status Register |
| =>lan8720 -c reset  => | 芯片软复位 |
| =>lan8720 -c down  => | 芯片设置掉电模式 |
| =>lan8720 -c loop  =>lan8720 -c loop -v 1  => | 芯片回路测试  芯片回路测试（远端） |

### can

别名：c

功能：控制can总线

语法： can [-c <command>] [-i <idx>] [-v <value>] [-e] [-r] <data>

options:

-c/--cmd info|baudrate|read|write|filter

-i/--idx index param

-v/--val value param

-e/--ext extended

-r/--rtr remote

data data args

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>can  error\_code: 0x0  error\_count: 0  => | 打印内部寄存器：  BCR: Basic Control Register  BSR: Basic Status Regir |

## 2、算法类

### flz

别名：无

功能：fasllz压缩算法

语法： flz [-d] <file>

options:

-d/--decompress decompress

file src -> dst

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>flz /s/test.dat  => | 压缩文件/s/test.dat |

### qlz

别名：无

功能：quicklz压缩算法

语法： qlz [-d] <file>

options:

-d/--decompress decompress

file src -> dst

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>qlz /s/test.dat  => | 压缩文件/s/test.dat |

## 3、LUA解释器

### lua

别名：无

功能：运行lua解释器

语法： lua [-v] [-r] [-s] [-c <command>] <file>

options:

-v/--version version

-r/--reset reset

-s/--stop stop

-c/--command command

file file name

|  |  |
| --- | --- |
| 示例 | 注释 |
| =>lua  > print(math.pi)  3.1415927  >  => | 运行lua解释器  Ctrl-C退出 |
| =>lua /s/test.lua  => | 运行解释器，执行脚本文件 |
| =>lua -r  => | 复位LUA虚拟机环境 |
| =>lua -s  => | 停止LUA虚拟机当前执行的程序 |
| =>lua -c “print(math.pi)”  3.1415927  => | 执行参数附带的短命令 |