W60X MicroPython User Guide

V0.1

Document History

Version	Completion Date	Revision Record	Author	Auditor
V0.1	2018-11-13	Initial release	LiLm	

Content

1   Introduction

1.1 Purpose

This document introduces how to use W60X under MicroPython.

1.2 Readers

The developers with W60X.

2   Introduction of MicroPython

MicroPython is a streamlined and effective realization with Python 3 language. It includes a small part of Python standard library and it has optimized the micro processer and limited environment.

Features of MicroPython on W60X:

- Support REPL (Python prompt) over UART0
- Support 16KB task stack and 45KB heap for running MicroPython
- Support most features of MicroPython and inter library (Unicode, high precision integer, single precision floating point number, complex number)
- Support hardware interface such as GPIO, UART, I^{2}C, PWM, WDT, TIMER, RTC and SPI.
- Support Wi-Fi network feature
- Support internal flash filesystem (27KB available)
- Support built-in FTP server transfer script files

The startup interface is in the following figure:

```
        \ \ \    / \/ \    / /
         \ \ \  /      \  / /
          \ \ \/        \/ /
           \ \  \      /  /
            \ \  \    /  /
             \ \  \  /  /
              \ \  \/  /
               \ \    /
               / /\  /\
              / /  \/  \
             / /   /\   \
            / /   /  \   \
           / /   /    \   \
          /_/   /      \   \
               /        \   \
              /__/    \__\

   WinnerMicro W600

MicroPython v1.10-279-g3c60627-dirty on 2019-05-24; WinnerMicro module with W600
Type "help()" for more information.
>>> help()
Welcome to MicroPython on the W600!

For generic online docs please visit http://docs.micropython.org/

For access to the hardware use the 'machine' module:

import machine
pb26 = machine.Pin(machine.Pin.PB_26, machine.Pin.OUT, machine.Pin.PULL_DOWN)
pb26.value(1)
pb27 = machine.Pin(machine.Pin.PB_27, machine.Pin.IN, machine.Pin.PULL_UP)
print(pb27.value())

Basic WiFi configuration:

import network
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.scan()                          # Scan for available access points
sta_if.connect("<AP_name>", "<password>") # Connect to an AP
sta_if.isconnected()                   # Check for successful connection

Control commands:
  CTRL-A        -- on a blank line, enter raw REPL mode
  CTRL-B        -- on a blank line, enter normal REPL mode
  CTRL-C        -- interrupt a running program
  CTRL-D        -- on a blank line, do a soft reset of the board
  CTRL-E        -- on a blank line, enter paste mode

For further help on a specific object, type help(obj)
For a list of available modules, type help('modules')
>>>
```

The usage method supported by MicroPython can be found in the docs directory.

Function modules can be downloaded from https://github.com/micropython/micropython-lib.

Users can download it and update to W60X chipset, then import it into the script.

3 Quick Start MicroPython

W60X has been transplanted successfully with MicroPython 1.10. It supports the source code package for users to rebuilt and also supports compiled firmware for update directly.

3.1 Firmware Download

The compiled firmware has been added on the website: http://www.winnermicro.com. Users can just download the firmware and update it to W60X.

3.2 Rebuild MicroPython

This operation is based on GCC building and can be used in shell directly. With Windows system, users should install Cygwin. The W60X_IDE includes Cygwin environment and is suggested to be used. Users can download the W60X_IDE from WinnerMicro's website.

3.2.1 Download Cross Compiling Tools

The GCC cross compiling tools used by W60X is arm-none-eabi-gcc, and the download address is:

https://launchpad.net/gcc-arm-embedded/4.9/4.9-2014-q4-major.

After decompression, the tools's path should be put into the environment variables. For example, put it into the directory /opt:

```
export PATH=$PATH:/opt/tools/arm-none-eabi-gcc/bin
```

This configuration can be write to ".bashrc" to make the change permanent.

3.2.2 Download WM_SDK Package

The SDK package can be downloaded from http://www.winnermicro.com .
Note: MicroPython is supported from version G3.1 with W60X SDK.

After decompressed, the environment variable "*WMSDK_PATH*" should be setted with the directory os WM_SDK. For example：

```
export WMSDK_PATH=/home/w60x/WM_SDK
```

This configuration can be write to ".bashrc" to make the change permanent.

There are some components in WM_SDK which is useless for MicroPython. These components can be cut off before compiling to reduce this code size. Users can open the WM_SDK/Include/wm_config.h and modify the

macro of such useless component from "CFG_ON" to "CFG_OFF".

The suggested useless components are:

```
#define TLS_CONFIG_HS_SPI        CFG_OFF
#define TLS_CONFIG_HOSTIF        CFG_OFF
#define TLS_CONFIG_RMMS          CFG_OFF
#define TLS_CONFIG_HTTP_CLIENT CFG_OFF
#define TLS_CONFIG_NTP           CFG_OFF
```

Note: During building, if it prompts the code size exceeding the ROM limit, the cutting off should be operated.

3.2.3   Download MicroPython

Please download the source code package from http://www.winnermicro.com and decompress it.

3.2.4   Compiling
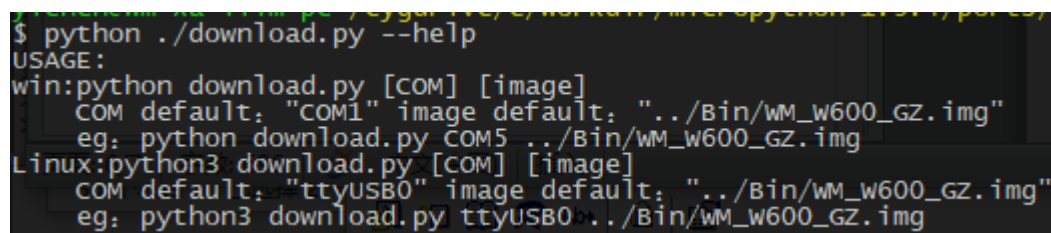
Entry the folder of ports/w60x of MicroPython project with shell command line, and operate the compiling command:

```
make V=s
```

After the compiling completed, the firmware will be generated under the folder of ports/w60x/build.

3.3 Update MicroPython

There is a python update script named "download.py" in ports/w60x/tools of MicroPython project. Input "python ./download.py --help" in shell command lines and view the using guide:

Users should set the port number by actual used and choose the firmware W60X_GZ.img to update.

Note: The size of W60X_GZ.img can't be larger than 352KB.

4 Examples for Command Lines

MicroPython supports interactive command line named REPL, the commands can be entered into the command line.

4.1 Basic Command of Printing

```
print('hello world')
print(b'bytes 1234\x01')
print(123456789)
for i in range(4):
    print(i)
```

4.2 Join the Wi-Fi

```
import network
sta_if = network.WLAN(network.STA_IF)
sta_if.active(True)
sta_if.scan()
sta_if.connect("WM2G","87654321")
sta_if.isconnected()
```

4.3 Create Socket

```
import socket
s=socket.socket()
addr=('192.168.18.92',1234)
s.connect(addr)
s.send("hello world!")
s.close()
```

4.4 Lighting up the LED Lamp

```
import machine
led=machine.Pin(machine.Pin.PB_26,machine.Pin.OUT,machine.Pin.PULL_FLOATING)
led.value(1)
led.value(0)
```

5 How to Use Script File

5.1 Convert Script File to Bytecode and Compile to Firmware

MicroPython supports compiling script file to firmware directly. After W60X is power on, the firmware will execute the _boot.by in the path ports/w60x/scrips defaultly. Users can put the Python script code into the file and let it can be executed automatically after power on.

All the files which are in the path ports/w60x/scrips will be compiled into the firmware. The command "pyexec_frozen_module" can be used to execute specified script file. For example:

```
pyexec_frozen_module("_boot.py");
```

Other script file can be called in _boot.py, both methods are OK.

Compling script file into firmware will not occupy the area of Flash file system, but will increase the size of image file. So users can choose the suitable method by the real situation.

5.2 Update Script File to W60X's Flash

5.2.1 How to Update Script File

W60X's internal Flash provides limited size of file system. Users can save the script file into the file system.

For convenience, the W60X MicroPython has integrated the feature of FTP server. After the W60X has joined the network, users can copy script file to W60X by FTP client on PC.

After W60X joined the network, following operation can be used to startup FTP server:

```
import w600
w600.run_ftpserver(port=21,username=None,password=None)
```

All the parameters have default values: default port number is 21; default user name and password are anonymous. User name and password should be string format with double quotation marks during setting.

When logging in with anonymous user name, users can only view and download files and can not upload, modify and delete files. If prompted for unauthorized operation, please set the user name and password and try again.

Following information will display when W60X FTP server starts up.

```
>>> import w600
>>> w600.run_ftpserver(port=21,username="root",password="123456")
ftpserver is running.
>>> ▉
```

Limited functions are supported by embedded FTP server and there are many kind of FTP clients, so there may be compatibility issues. If users can not operate, they can try to modify FTP connecting mode (active/passive).

The firewall (iptables) should be disable in Linux system, otherwise, some host PCs will not connect to FTP.

Following is the example with FileZilla:
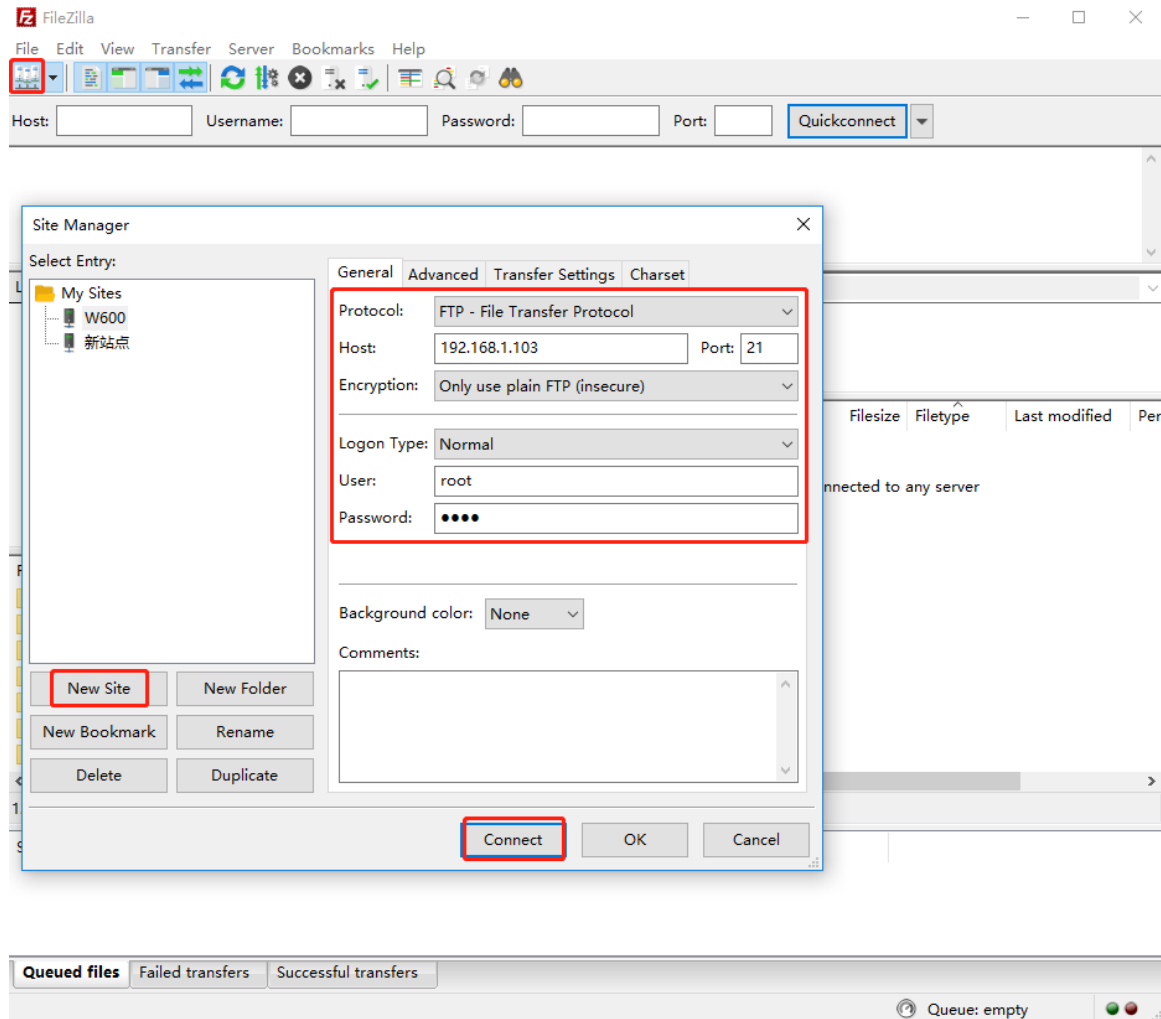
Following is the operation for setting active or passive mode:
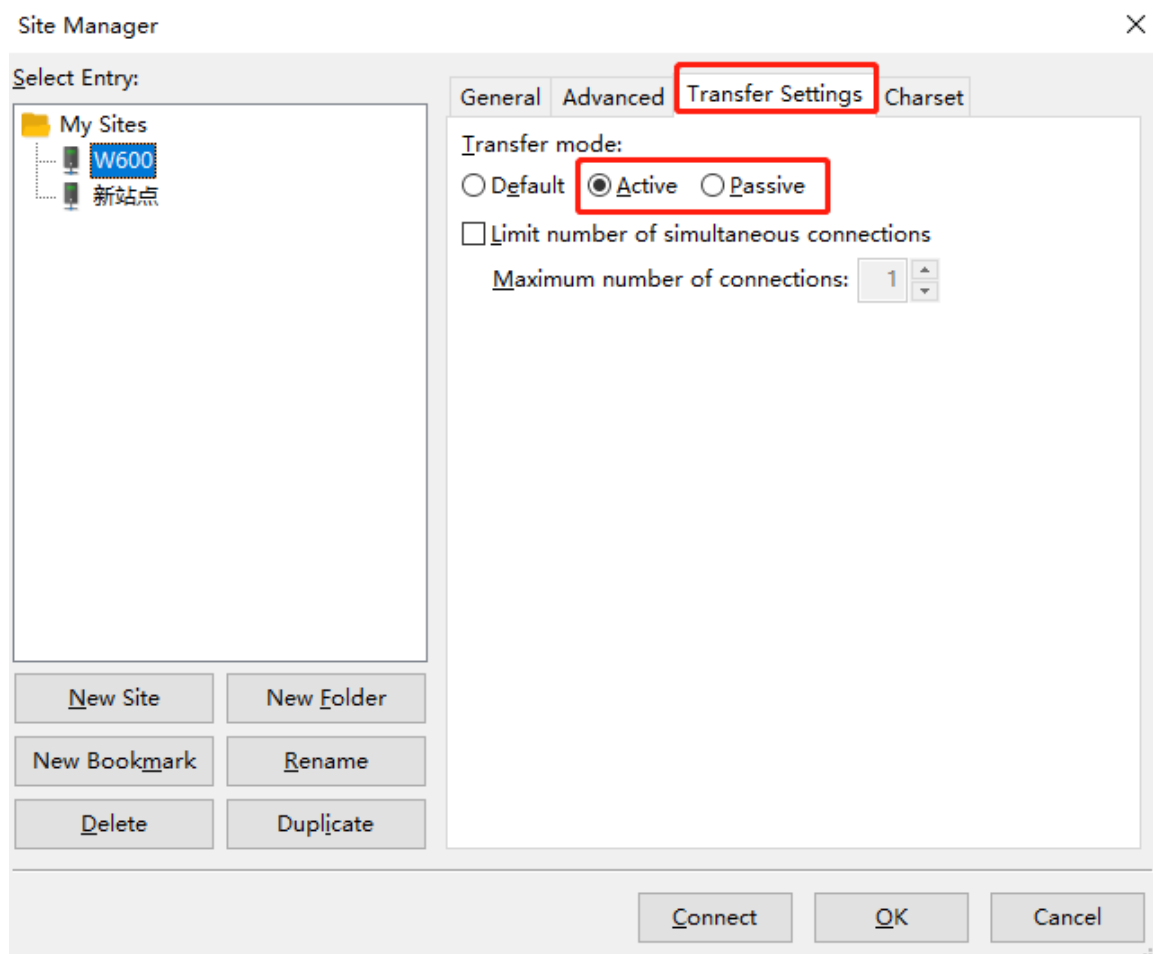
5.2.2 Flash File Structure

There are following files in W60X's default file system:

These files are created defaultly by system, users can modify them directly.

When W60X powers on, the boot.py script will be executed firstly. And then the main.py will be executed. If some user code will be executed, such code can be writen into these two script files. The initial code can be

writen into boot.py and function code can be writen into main.py. Users can add new script files according to their own requirement.

The script file easyw600.py is integrated in W60X, and this script file supports some common functions. Following is some operation examples:

```
import easyw600
easyw600.scan() : scan Wi-Fi APs nearby W60X.
easyw600.oneshot() : start up oneshot configuration. After joined the network successfully, the IP
address can be printed out.
easyw600.connect(ssid="myssid", password=None) : start to connecting the Wi-Fi, the IP address can be
  printed out after joined the network successfully.
easyw600.disconnect() : disconnect the network.
easyw600.createap(ssid="w60x_softap", password=None) : create a soft AP.
easyw600.closeap() : close soft AP.
easyw600.ftpserver() : start up integrated FTP server, the default port number is 21, the user name is
root, the password is root.
```

The folders such as "cert", "lib" and "sys" are reserved for extended functions. After users have downloaded new function modules, these modules can be put into such folders. Then these modules can be used in scripts.

6   Version Specification

Current W60X's MicroPython version number is W60X_MicroPython_1.10_B1.1. Following is the transplanted modules:

名称

- _thread.rst
- array.rst
- builtins.rst
- cmath.rst
- framebuf.rst
- gc.rst
- machine.I2C.rst
- machine.Pin.rst
- machine.rst
- machine.RTC.rst
- machine.Signal.rst
- machine.SPI.rst
- machine.Timer.rst
- machine.UART.rst
- machine.WDT.rst
- math.rst
- micropython.rst
- network.rst
- sys.rst
- ubinascii.rst
- ucollections.rst
- uctypes.rst
- uerrno.rst
- uhashlib.rst
- uheapq.rst
- uio.rst
- ujson.rst
- uos.rst
- ure.rst
- uselect.rst
- usocket.rst
- ustruct.rst
- utime.rst
- uzlib.rst

SSL module is not supported in this version, and it will be solved by later updates. Please look forward to it.