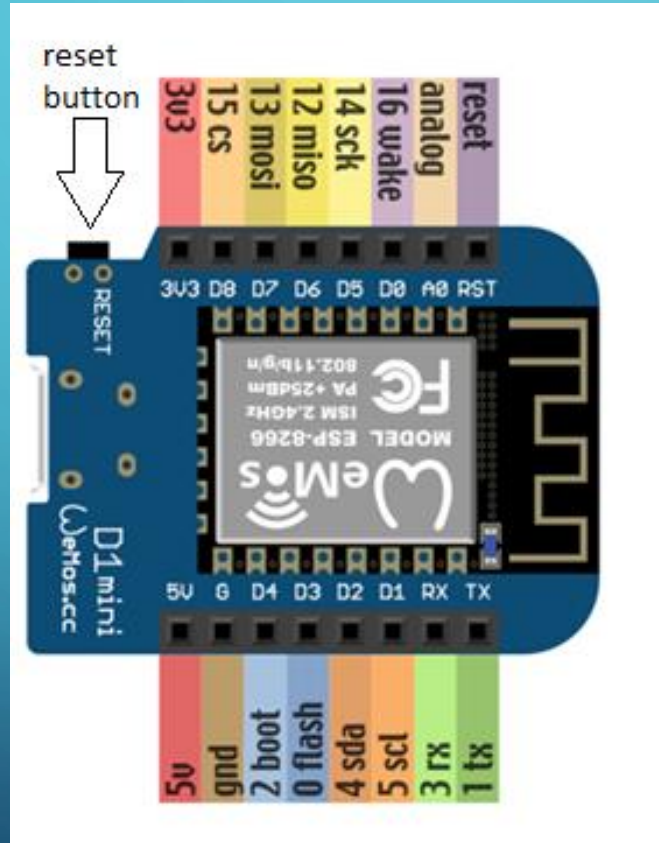


A decorative graphic on the left side of the slide, consisting of a network of light blue lines and small circles, resembling a circuit board or a stylized tree structure, extending from the top to the bottom of the frame.

# INTRO TO THE ESP8266 MICROCONTROLLER AND MICROPYTHON

BY M. ALLEMANG

# THE D1 MINI ESP8266



## FEATURES:

- 11 Digital I/O most with PWM capability
- 3.3 Volt digital I/O only...not 5V TTL
- 1 analog input
- Connects via USB – COM port
- Obtains its power via USB

## FEATURES(CONT):

- Programs written in python
- Stored in Non Volatile memory (EEPROM)
- Has networking capability such as WIFI (access point or client) & TCP/IP stack

# PROGRAM DEVELOPMENT

- Interact with the board from host computer using its REPL and comm app such as TeraTerm
  - Not very useful... ok for testing the board or it's peripherals
- Write .py program on host computer using ~~IDLE or~~ Notepad++, then copy .py file to the board and “import” it using the REPL
- Name your program ‘main.py’ and it will run automatically at startup (only do so once program is debugged!!!)

# PROGRAM DEVELOPMENT(CONT)

- We will use a windows/linux program called *rshell* to interact with the board, copy programs between host computer and the board, run programs etc.
- The board can be “bricked” by using *main.py* and configuring the RX circuit/pin as an OUTPUT. In this case I am able to re-flash micropython to the board... like setting it back to factory default (resetting your phone).

# PROGRAM DEVELOPMENT(CONT)

- No way to single step programs, no breakpoints.
- Debug your programs by using strategic *print* statement
- DO NOT USE ANY SPACES IN YOUR PROGRAM FILENAMES
- DO NO USE ANY DASHES '-' IN YOUR PROGRAM FILENAMES
- DO NOT NAME YOUR FILES THE



# PROGRAM DEVELOPMENT(CONT)

- Typical Program Development scenario:

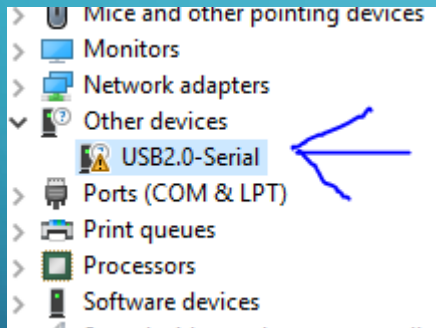
- 1) Edit program using Notepad++ and save it as `myprog.py`
- 2) Using *rshell* copy the file to the board (you will need to learn some basic UNIX/Linux commands: `cp` – copy, `rm` – remove/delete, `ls` – list files, `cat` – display content)
- 3) Using *rshell* enter into the REPL
- 4) `import myprog` to run your program
- 5) Go back to step 1 to fix syntax and logic errors



# LET'S GIVE IT A TRY

## 1) GET SOFTWARE READY IN ROOM B1055

- Plug your board into protoboard and connect USB
- Using Device manager – setup com port




- Right click and update driver -> search online automatically
- Once driver is good to go, RECORD YOUR COM PORT NUMBER!!

## 2) WRITE A PROGRAM

- Using Notepad++ write a program that displays the message “hello world” and save it (NAME IT **hello.py** ) on your S: drive (you may change the default app for .py files to be notepad++ instead of Python depending on your preference)
- As soon as you save it as a .py file, the keywords should change colour

### 3) USE RSHELL TO COMMUNICATE WITH THE BOARD

- Using windows explorer,  right click on your course folder on S: drive while holding the SHIFT key and choose “open command prompt here” or “open powershell here”
- Type **rshell** and tap ENTER
- Type **connect serial comx** where x is your com port number (make a mental note of typical board response)
- Type **ls** and you should see your .py file
- Type **ls /pyboard** and you should see the files on your board

## 4) ENTER AND EXIT THE REPL

- Type `repl` (I'm not going to tell you to tap ENTER anymore)
- Notice the colour change, type `print(5+3)`
- How do you get out of the repl ? . . . .MAKE IT SO!

## 5) COPY YOUR PROGRAM TO YOUR BOARD

- The copy command is: `cp <from> <to>`
- TYPE `cp hello.py /pyboard` this means copy the file named hello.py from your default directory(folder) which is your s: drive folder, to your esp8266 (known as /pyboard)
- TYPE `cp /pyboard/hello.py helloagain.py` this means copy hello.py from the pyboard to your default directory (which is your S: drive folder) and name it helloagain.py      Get it?

# LETS PRACTICE

## (I FEEL A PRACTICAL TEST IS IN THE NEAR FUTURE)

- LIST THE FILES ON YOUR `pyboard` USING `ls`
- LIST THE FILES IN YOUR DEFAULT DIRECTORY USING `ls`
- DISPLAY THE CONTENTS (`cat`) OF YOUR `hello.py` FILE
- COPY `boot.py` FROM YOUR `/pyboard` TO YOUR DEFAULT DIRECTORY
- EXAMINE THE CONTENTS OF `boot.py` (just for fun)

# RUN YOUR PROGRAM

- TO RUN YOUR PROGRAM `hello.py` YOU HAVE TO:
- 1) BE IN THE REPL
- 2) TYPE `import hello.py`
- 3) after your program exits, you might have to CTRL-D to do a soft reboot to be able to run your program again



# QUESTIONS ?

- AT HOME YOU CAN INSTALL RSHELL BY INSTALLING PYTHON VERSION 3.X
- (MAKE SURE YOU INCLUDE PIP IN THE PYTHON INSTALLATION OPTIONS)
- In a command prompt/power shell type:

**pip install rshell**