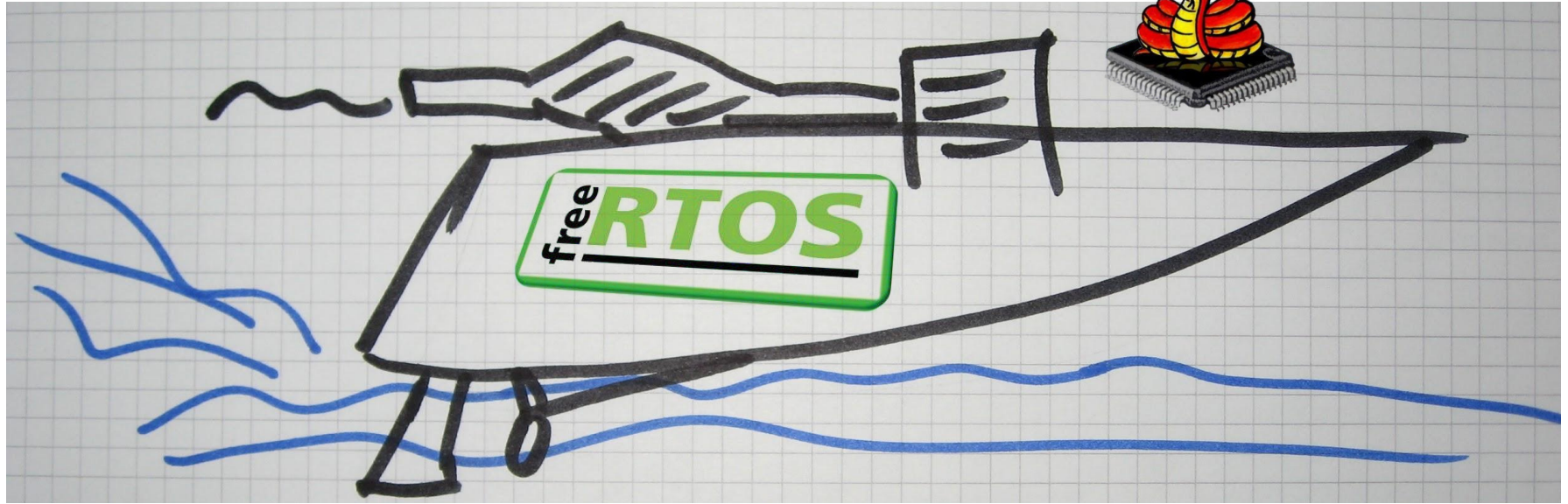


MICROPYTHON ON THE WATER

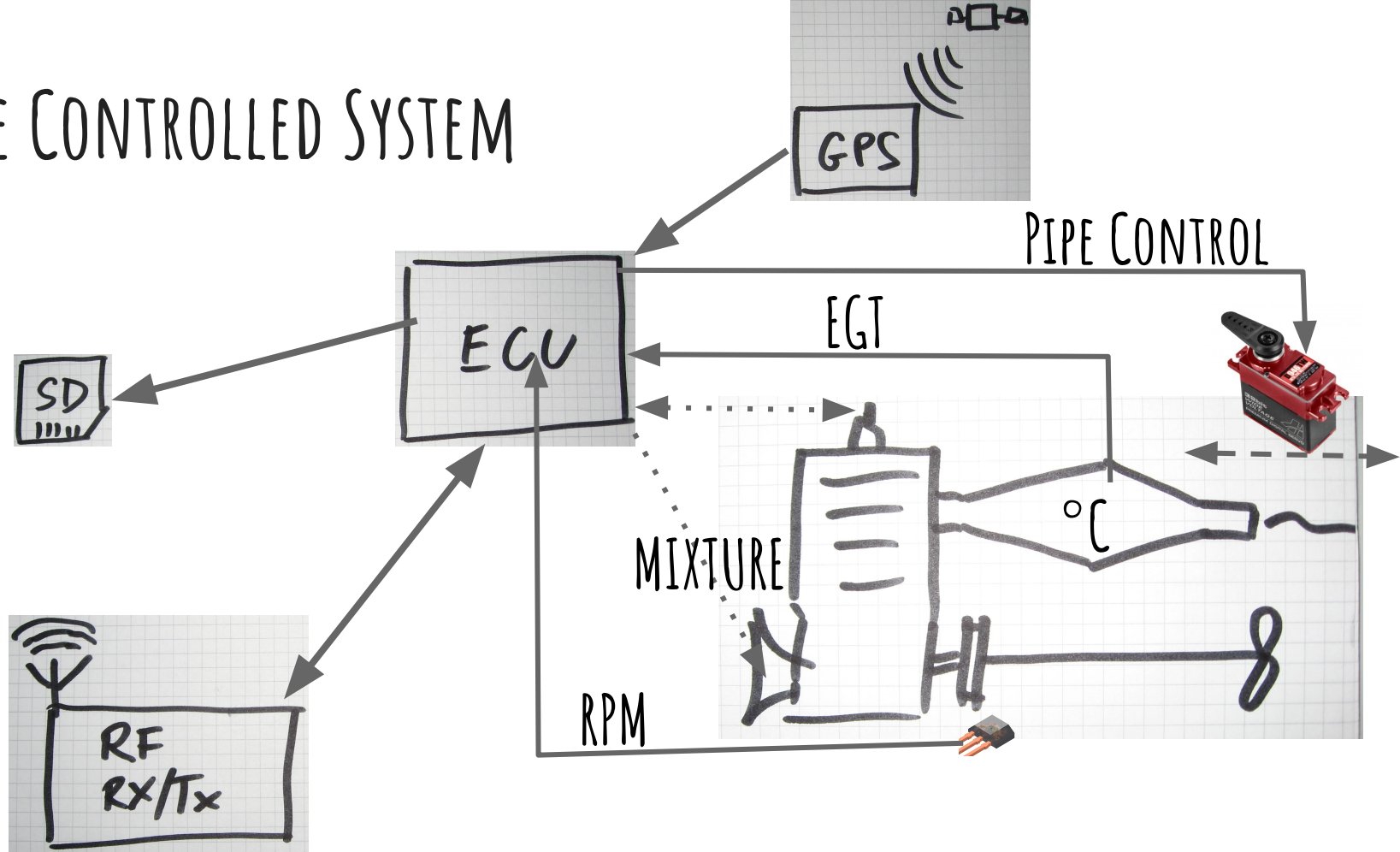
PyCon CZ 2017, 8. 6. 2017

jan.capek@{brains.cz,slushpool.com}

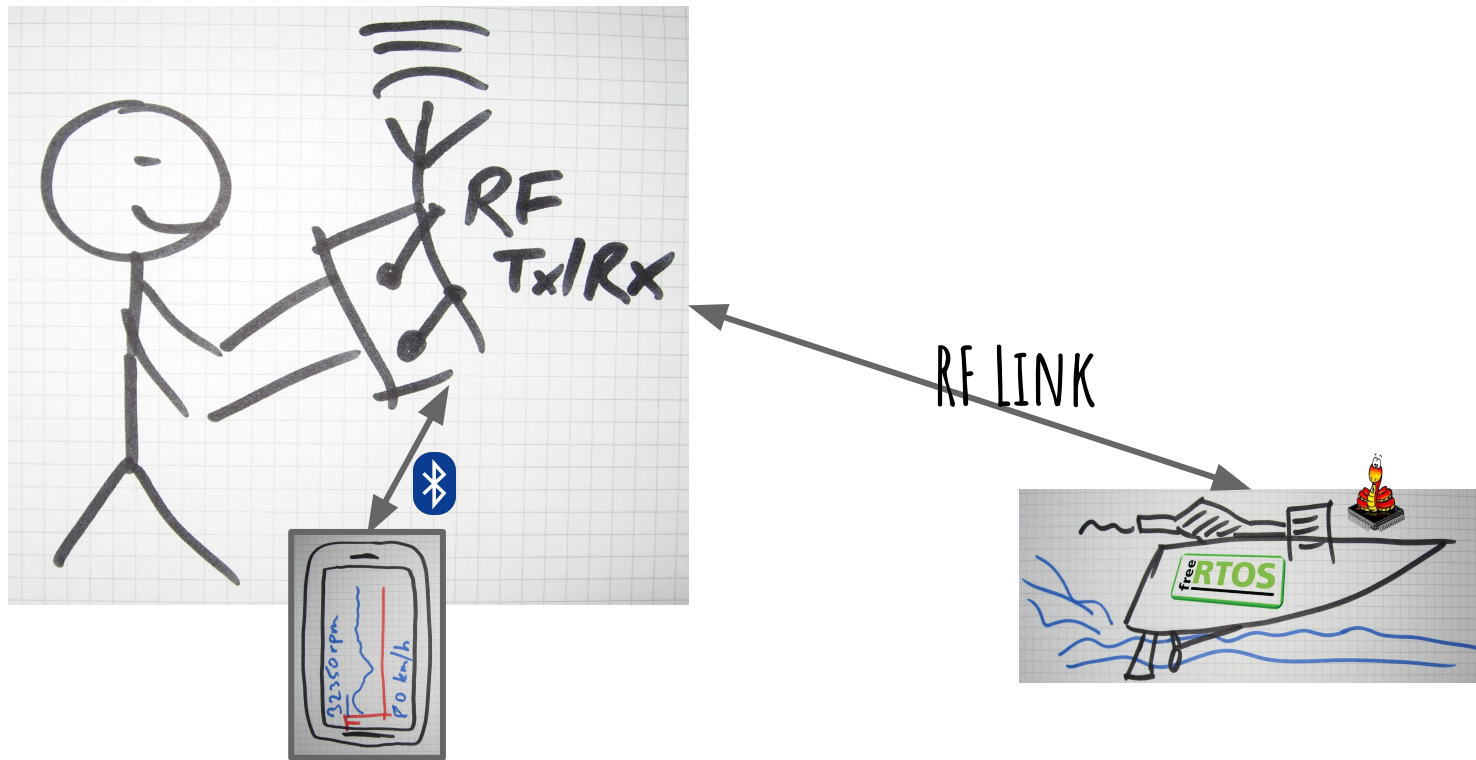
THE POWERFUL DUO



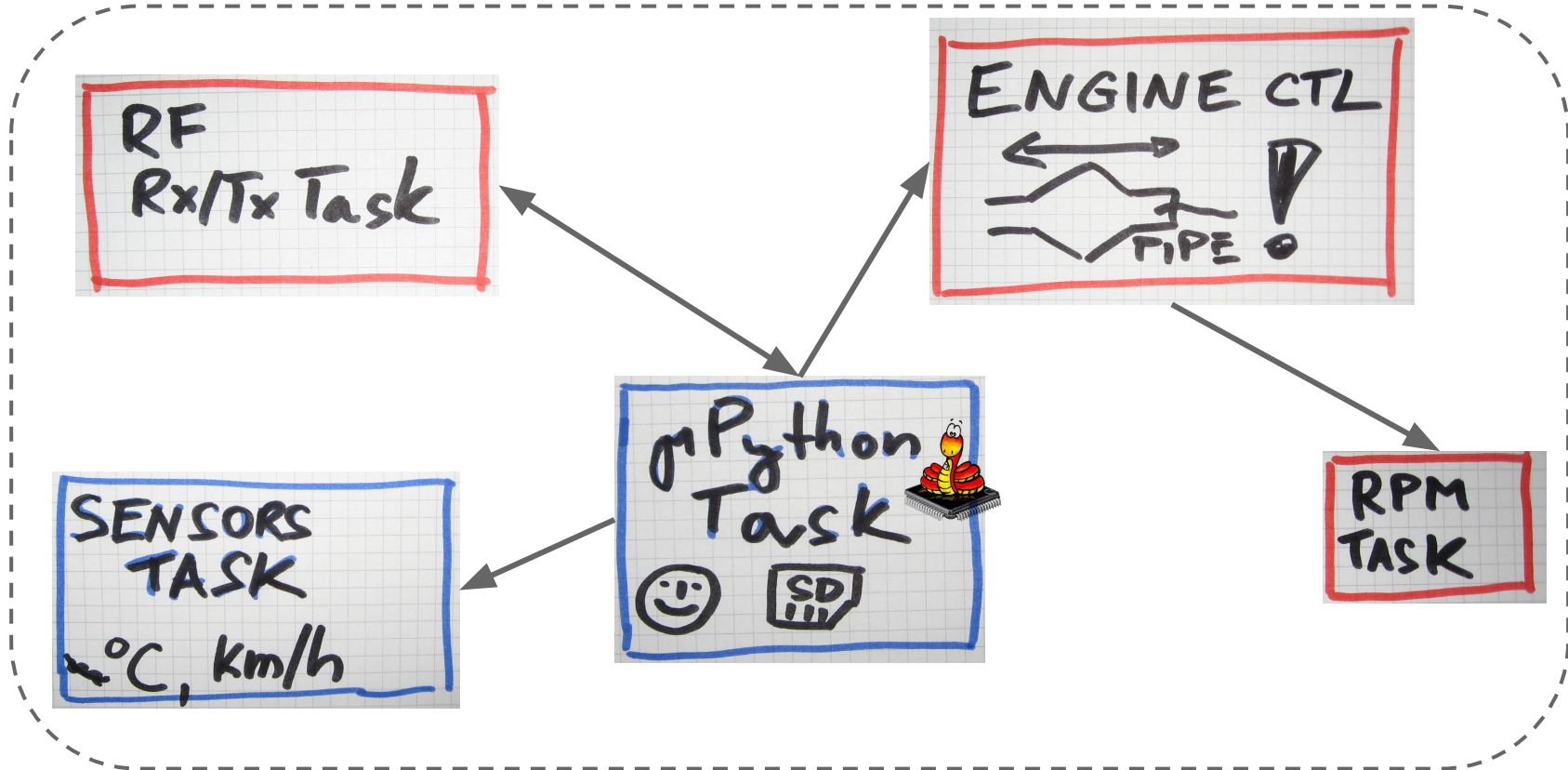
THE CONTROLLED SYSTEM



THE BOAT DRIVER



RTOS



EXAMPLE

```
def pipe_init(app):
    conf_path = '%s/pipe.cfg' % mount_path
    with uio.open(conf_path, 'r') as conf:
        content = conf.readlines()

    pipe_ctl_channel = int(content[0])
    full_throttle_threshold = int(content[1])
    app.pipe_ctl_channel(pipe_ctl_channel)
    app.full_throttle_threshold(full_throttle_threshold)

    print('Reading tuned pipe configuration')
    i = 0
    for line in content[2:]:
        pipe_setting = [i] + [int(x) for x in line.split()]
        print('idx: %s rpm: %s length: %s' % (pipe_setting[0],
                                                pipe_setting[1],
                                                pipe_setting[2]))

        app.pipe_table_entry(pipe_setting)
        i += 1
```


EXAMPLE

```
def pipe_init(app):
    conf_path = '%s/pipe.cfg' % mount_path
    with uio.open(conf_path, 'r') as conf:
        content = conf.readlines()

    pipe_ctl_channel = int(content[0])
    full_throttle_threshold = int(content[1])
    app.pipe_ctl_channel(pipe_ctl_channel)
    app.full_throttle_threshold(full_throttle_threshold)

    print('Reading tuned pipe configuration')
    i = 0
    for line in content[2:]:
        pipe_setting = [i] + [int(x) for x in line.split()]
        print('idx: %s rpm: %s length: %s' % (pipe_setting[0],
                                                pipe_setting[1],
                                                pipe_setting[2]))

        app.pipe_table_entry(pipe_setting)
        i += 1
```

EXAMPLE

```
def pipe_init(app):
    conf_path = '%s/pipe.cfg' % mount_path
    with uio.open(conf_path, 'r') as conf:
        content = conf.readlines()

    pipe_ctl_channel = int(content[0])
    full_throttle_threshold = int(content[1])
    app.pipe_ctl_channel(pipe_ctl_channel)
    app.full_throttle_threshold(full_throttle_threshold)

    print('Reading tuned pipe configuration')
    i = 0
    for line in content[2:]:
        pipe_setting = [i] + [int(x) for x in line.split()]
        print('idx: %s rpm: %s length: %s' % (pipe_setting[0],
                                                pipe_setting[1],
                                                pipe_setting[2]))

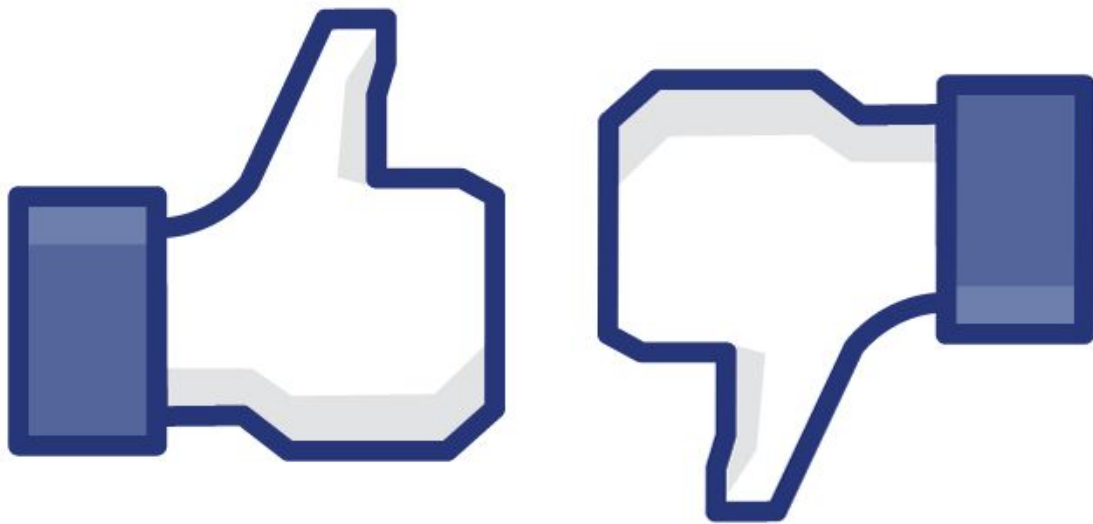
        app.pipe_table_entry(pipe_setting)
        i += 1
```


WHAT ABOUT ASYNC PROGRAMMING?



RTOS AWARE
EVENTLOOP

WHERE ARE WE NOW WITH UPY?



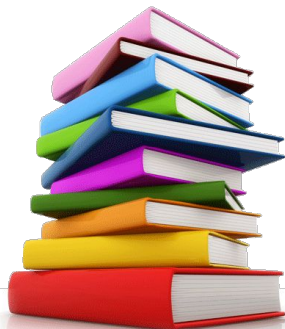
WHO SHOULD BUILD THIS?



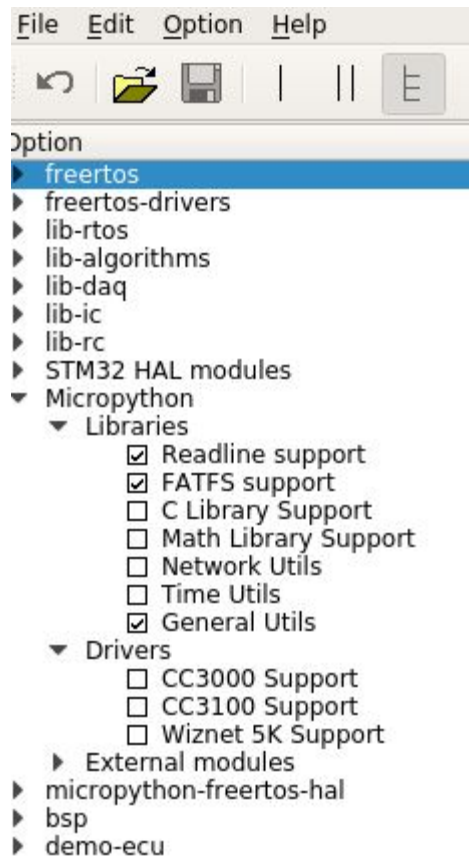
+



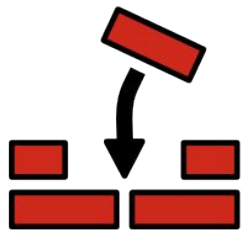
+



+



THE CARPENTER'S BUILDTOOL



SCONS

+



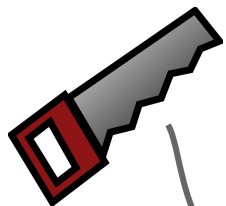
LINUX KERNEL BUILD SYSTEM

```
objs-y += \  
    moductypes.o \  
    modujson.o \  
    modure.o \  
    moduzlib.o \  
    moduheapq.o \  
    moduhashlib.o \  
    modubinascci.o \  
    machine_mem.o \  
    modussl.o \  
    modurandom.o \  
    modwebsocket.o \  
    fsusermount.o \  
    vfs_fat.o \  
    vfs_fat_ffconf.o \  
    vfs_fat_diskio.o \  
    vfs_fat_file.o \  
    vfs_fat_lexer.o \  
    vfs_fat_misc.o \  
    moduos_dupterm.c
```

```
objs-$(CONFIG_MICROPYTHON_EXTMOD_PY_USSL) += modussl.o
```

SCONSCRIPT WITH PILA

```
1  Import('env')
2
3  env.QstrFeatureObject(source=[ 'moductypes.c',
4                                'modujson.c',
5                                'modure.c',
6                                'moduzlib.c',
7                                'moduheapq.c',
8                                'moduhashlib.c',
9                                'modubinascii.c',
10                               'machine_mem.c',
11                               'modussl.c',
12                               'modurandom.c',
13                               'modwebsocket.c',
14                               'fsusermount.c',
15                               'vfs_fat.c',
16                               'vfs_fat_ffconf.c',
17                               'vfs_fat_diskio.c',
18                               'vfs_fat_file.c',
19                               'vfs_fat_lexer.c',
20                               'vfs_fat_misc.c',
21                               'moduos_dupterm.c'])
22
23  env.QstrFeatureObject(source=[ 'modussl.c'],
24                             is_enabled=env['CONFIG'].MICROPYTHON_EXTMOD_PY_USSL)
```



KCONFIG EXAMPLE

```
menu "External modules"
config MICROPYTHON_EXTMOD_PY_USSL
    bool "MicroSSL module support"
    default n
    ---help---
        Micropython module for uSSL library
endmenu
```



[HTTPS://BRAIINS.GITHUB.IO/PYCON2017-MICROPYTHON-ON-THE-WATER-NOTES/](https://BRAIINS.GITHUB.IO/PYCON2017-MICROPYTHON-ON-THE-WATER-NOTES/)

