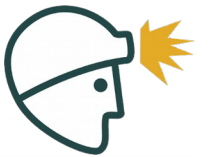


Introduction



WSCF 2019
CYBER PHYSICAL SYSTEMS WORKSHOP





Software

- Whitecat IDE
 - Lua RTOS
 - Blocks Language
- ZeroBrane Studio
- Terminal emulator program
- Whitecat Console (wcc, command line tool).



What's Lua RTOS?

- Lua RTOS is a real-time operating system.
- Minimal requirements of memory.
- Available for ESP32, ESP8266 and PIC32MZ.

Lua interpreter					
PIO	ADC	CAN	PWM	LoRa	MQTT
SPI	I2C	UART	...	Sensors	...
Real-Time micro-kernel					
Hardware Abstraction Layer					

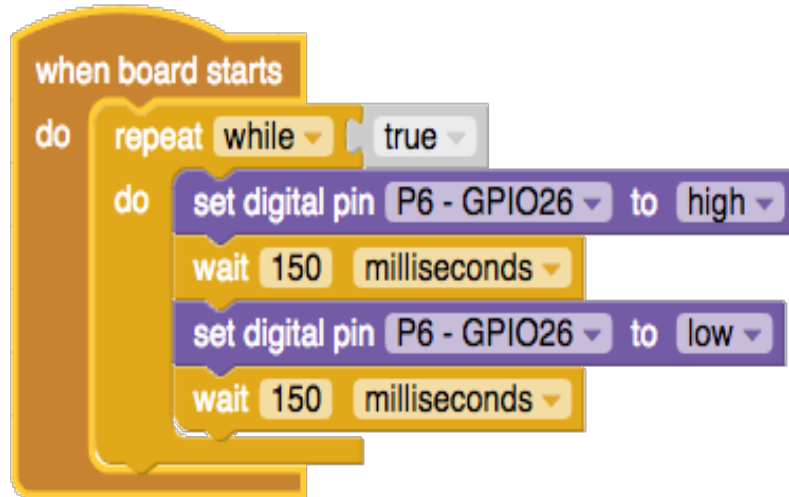


Lua Threads

- Capacity to execute Lua functions concurrently.
- Shares the same Lua state.
- Makes Lua RTOS programs more natural and intelligible



Hello World



```
thread.start(function()  
  pio.pin.setdir(pio.OUTPUT, pio.GPIO26)  
  pio.pin.setpull(pio.NOPULL, pio.GPIO26)  
  
  while true do  
    pio.pin.setval(1, pio.GPIO26)  
    tmr.delayms(150)  
    pio.pin.setval(0, pio.GPIO26)  
    tmr.delayms(150)  
  end  
end)
```



Get the Lua RTOS firmware

- Flash precompiled firmware (using wcc).
- Build by yourself (from git repo)



Whitecat Console

- Command line tool
- Allows the programmer to flash a Lua RTOS

```
$ wcc
wcc -p port | -ports [-ls path |
    [-down source destination] | [-up source destination] |
    [-f | -ffs] | [-erase] | -d]

-ports:      list all available serial ports on your computer
-p port:     serial port device, for example /dev/tty.SLAB_USBtoUART
-ls path:    list files present in path
-down src dst: transfer the source file (board) to destination file (computer)
-up src dst:  transfer the source file (computer) to destination file (board)
-f:          flash board with last firmware
-ffs:        flash board with last filesystem
-erase:      erase flash board
-d:          show debug messages
```



Connect to the console

- Connect to the Lua RTOS console using your favorite terminal emulator program.
- Connection parameters are:
 - speed: 115200 bauds
 - data bits: 8
 - stop bits: 1
 - parity: none
 - terminal emulation: VT100



Example using picocom

- Connect the board to the computer.
- Check serial port

```
$ wcc -ports
```

- Connect to the cosole

```
$ picocom --baud 115200 /dev/ttyUSB0
```



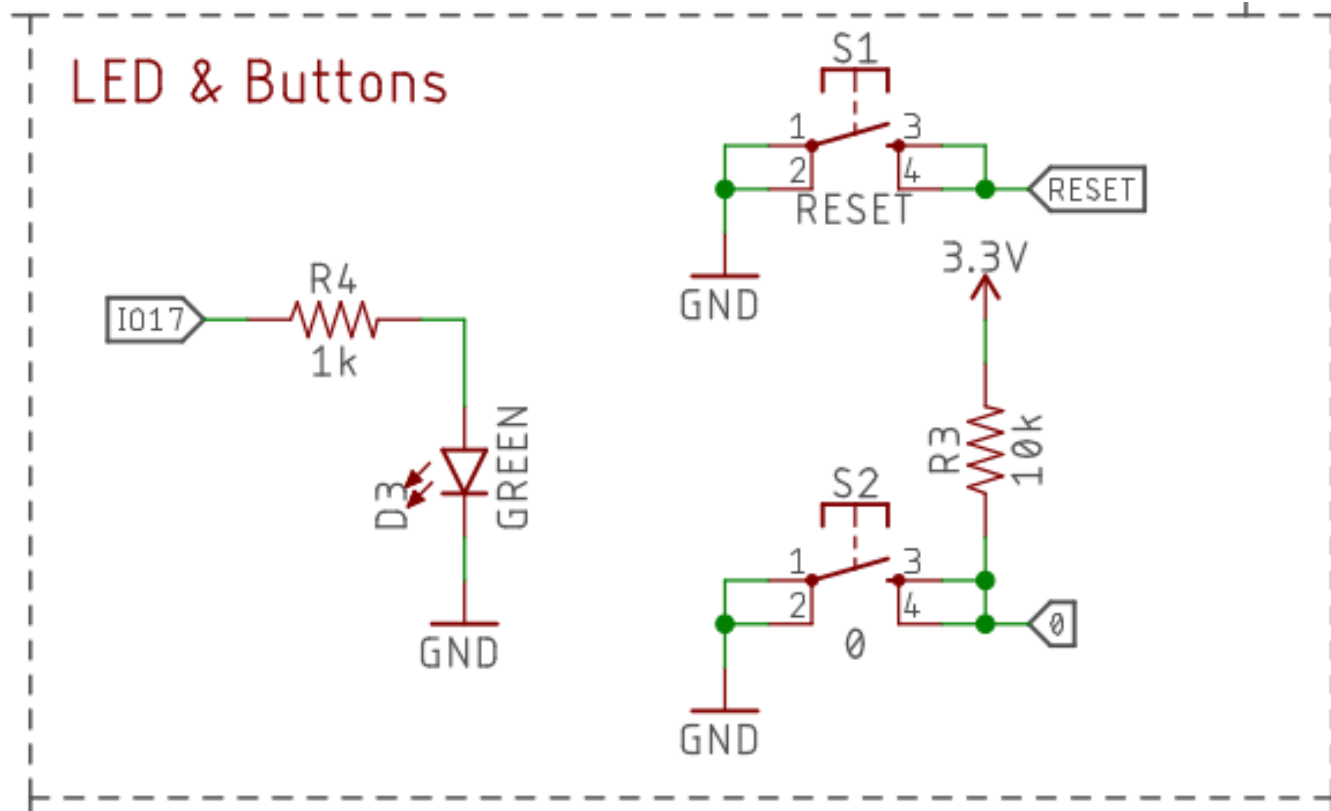
Exercise

- Schematics
- First I/O program
 - Transfer script from the host to the target.
 - Run script

```
/> dofile('first-io')
```
- Change the led state every time the button is pressed.



Exercise





Questions