Cypress IoT-AdvantEdge™ Webinar Series

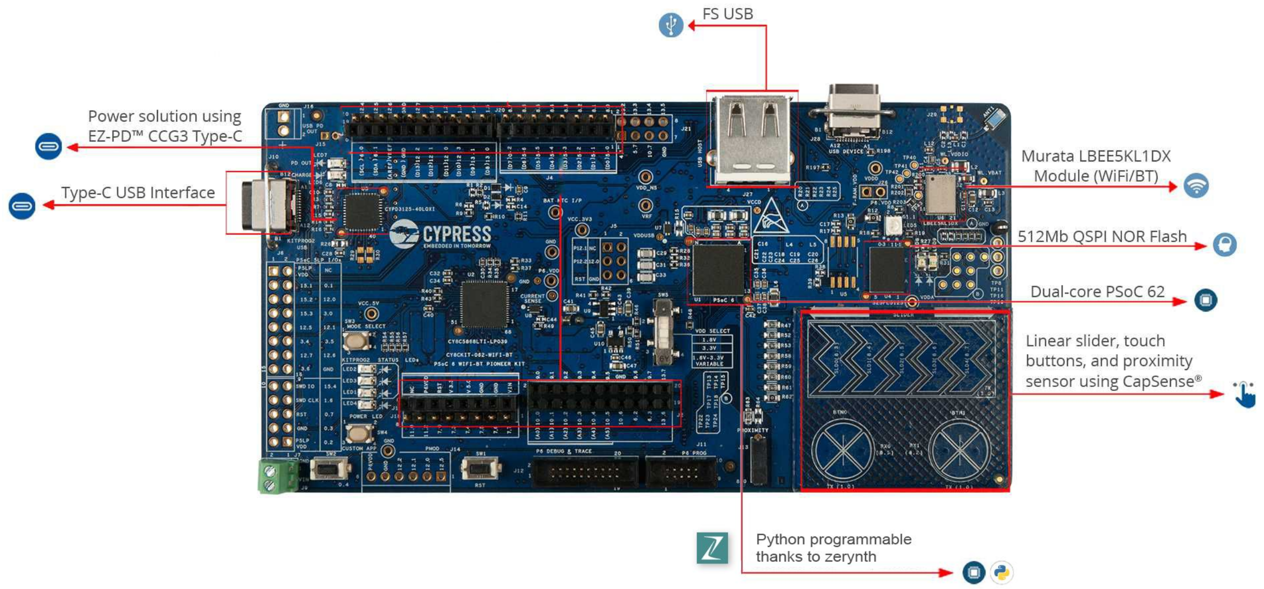
Python on PSoC® 6 MCUs for IoT and Blockchain Applications

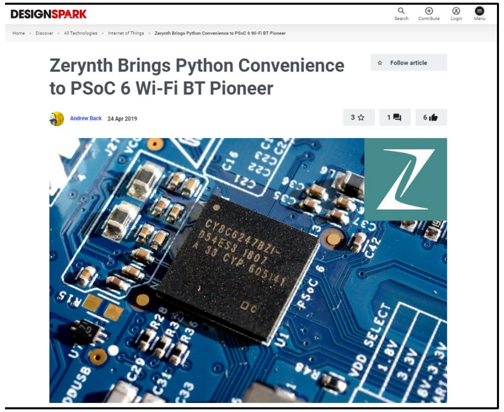
Jul 16, 2020

Giacomo Baldi
Co-Founder | CTO
Zerynth®

Zerynth OS for PSoC 6

Zerynth OS has been ported on PSoC 6 in cooperation with RS Components


FS USB
Power solution using EZ-PD™ CCG3 Type-C
Type-C USB Interface
Murata LBEE5KL1DX Module (WiFi/BT)
512Mb QSPI NOR Flash
Dual-core PSoC 62
Linear slider, touch buttons, and proximity sensor using CapSense®
Python programmable thanks to zerynth
CYPRESS



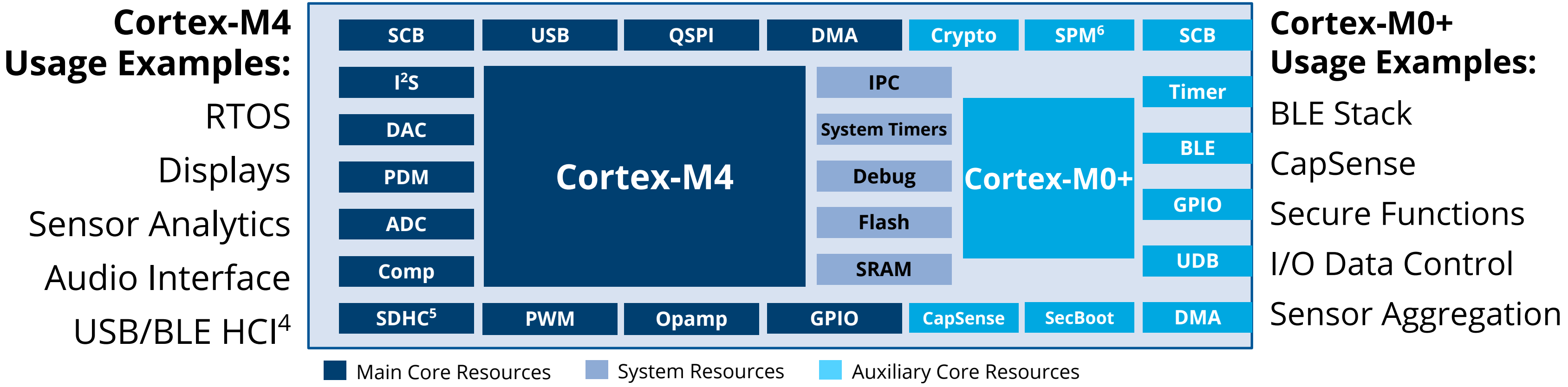
Zerynth Brings Python Convenience to PSoC 6 Wi-Fi BT Pioneer

More info: https://www.rs-online.com/designspark/zerynth-brings-python-convenience-to-psoc-6-wi-fi-bt-pioneer

PSoC 6 - Industry's Most Flexible MCU Architecture for the IoT

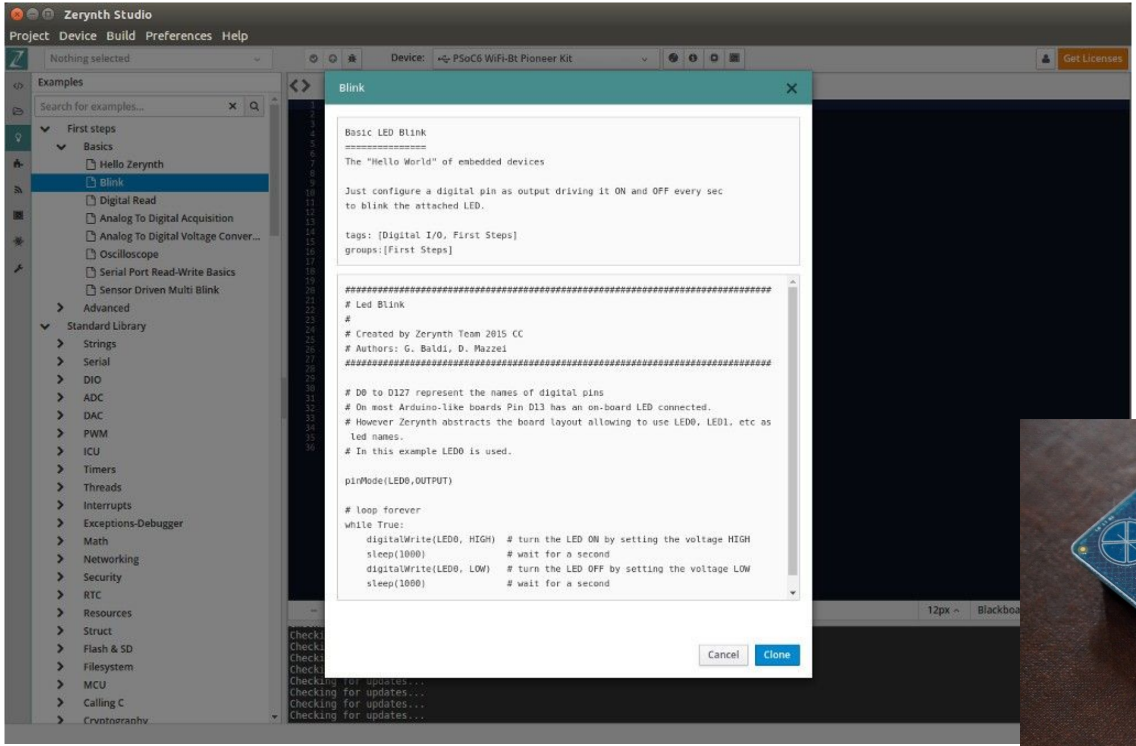
- Multiple wired and wireless connectivity options such as BLE[1], Wi-Fi[2], and USB to support Internet, cloud-based services
- Software-defined peripherals to create custom AFEs[3] and to support last-minute design changes while minimizing PCB re-spins
- CapSense, the industry's best capacitive sensing solution, to support sleek, next-generation user-interfaces
- Flexible dual-core architecture to optimize system power consumption and performance

PSoC 6 Dual-Core MCU Architecture



[1] Bluetooth Low Energy
[2] PSoC 6 as host MCU with Cypress' wireless radio products (WICED)
[3] Analog front ends
[4] Host Controller Interface
[5] Secure Digital High Capacity
[6] Arm-based SPM available in PSoC 64 line

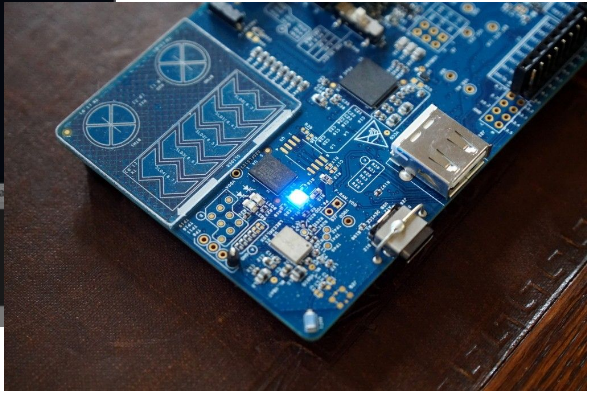
Python on PSoC 6 - Hello World



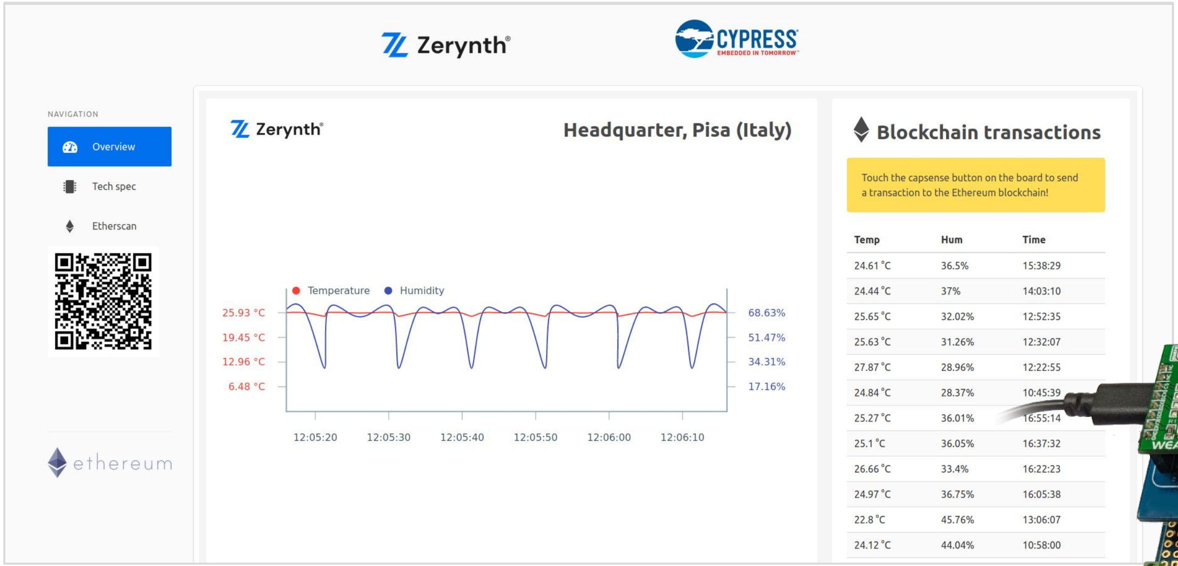
```
Basic LED Blink
===============
The "Hello World" of embedded devices

Just configure a digital pin as output driving it ON and OFF every sec
to blink the attached LED.

tags: [Digital I/O, First Steps]
groups:[First Steps]
```

```
################################################################################
# Led Blink
#
# Created by Zerynth Team 2015 CC
# Authors: G. Baldi, D. Mazzei
################################################################################

# D0 to D127 represent the names of digital pins
# On most Arduino-like boards Pin D13 has an on-board LED connected.
# However Zerynth abstracts the board layout allowing to use LED0, LED1, etc as
 led names.
# In this example LED0 is used.

pinMode(LED0,OUTPUT)

# loop forever
while True:
    digitalWrite(LED0, HIGH)  # turn the LED ON by setting the voltage HIGH
    sleep(1000)               # wait for a second
    digitalWrite(LED0, LOW)   # turn the LED OFF by setting the voltage LOW
    sleep(1000)               # wait for a second
```

```
Checking for updates...
Checking for updates...
Checking for updates...
Checking for updates...
```



Hello Blockchain: Ethereum IoT demo

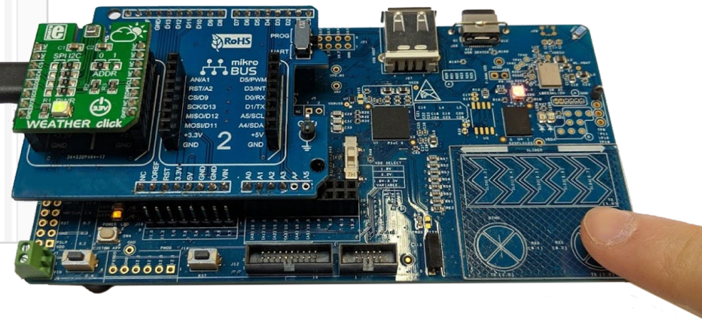


Headquarter, Pisa (Italy)

Blockchain transactions

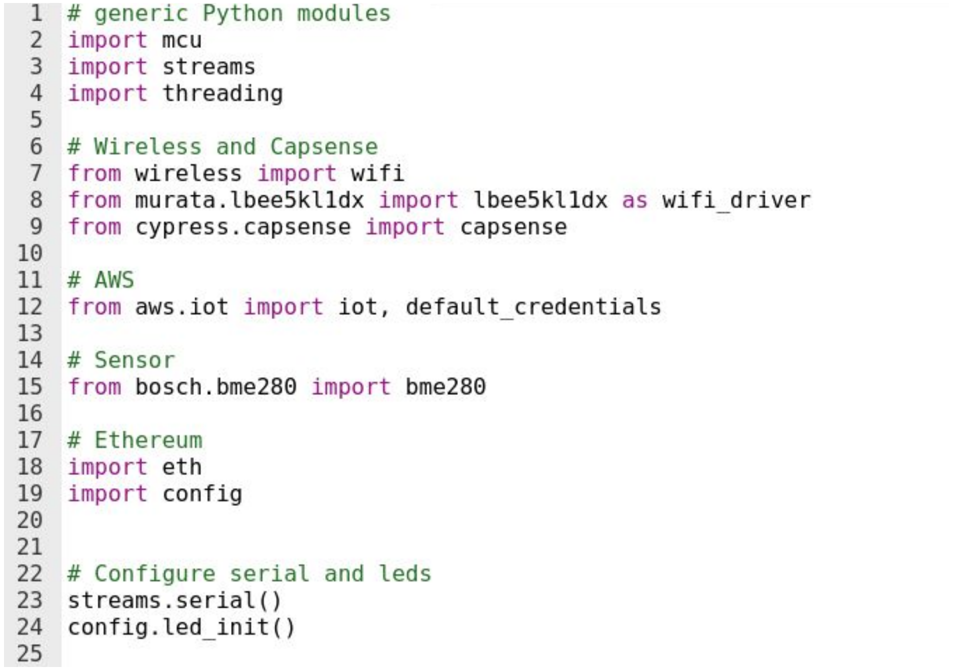
Touch the capsense button on the board to send a transaction to the Ethereum blockchain!

Temp	Hum	Time
24.61 °C	36.5%	15:38:29
24.44 °C	37%	14:03:10
25.65 °C	32.02%	12:52:35
25.63 °C	31.26%	12:32:07
27.87 °C	28.96%	12:22:55
24.84 °C	28.37%	10:45:39
25.27 °C	36.01%	16:55:14
25.1 °C	36.05%	16:37:32
26.66 °C	33.4%	16:22:23
24.97 °C	36.75%	16:05:38
22.8 °C	45.76%	13:06:07
24.12 °C	44.04%	10:58:00



https://psoc6.demo.zerynth.com

Hello Blockchain: Ethereum IoT demo

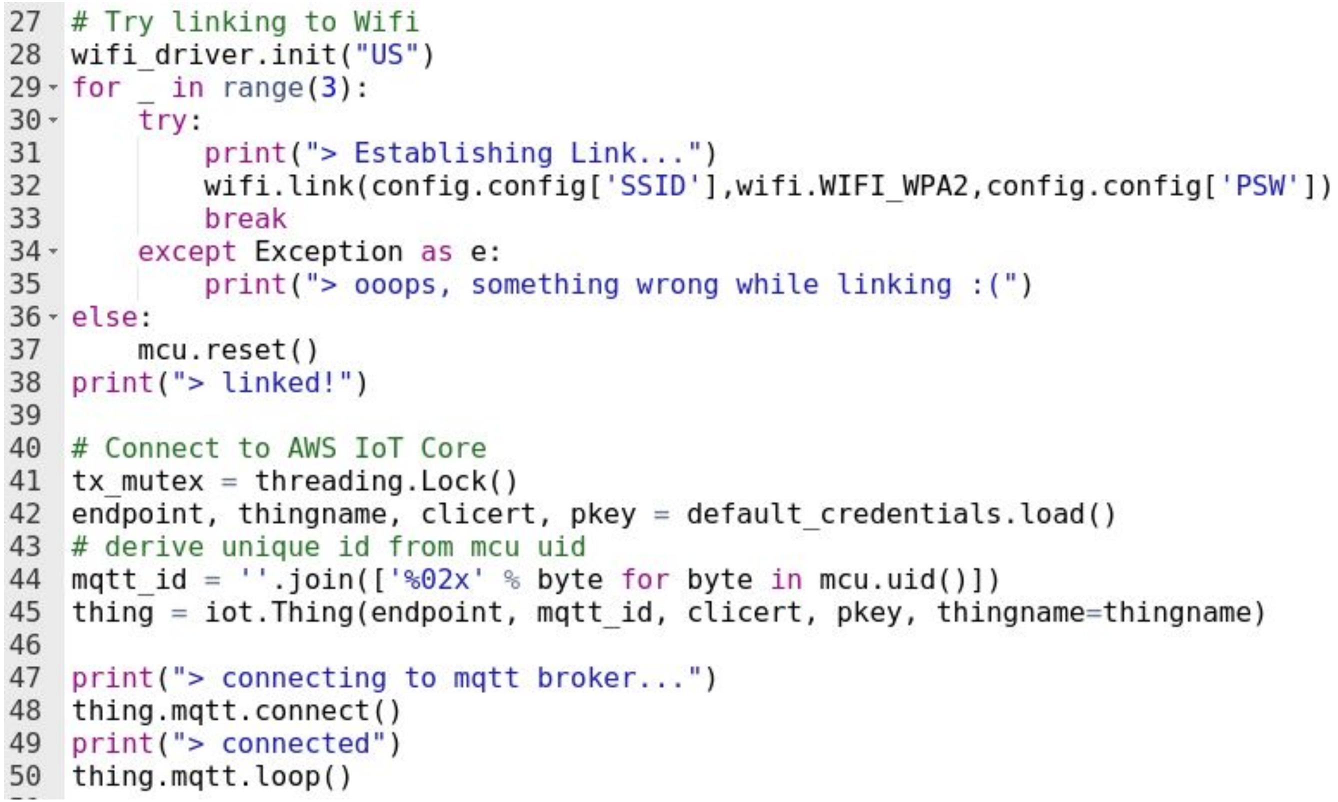


```python
# generic Python modules
import mcu
import streams
import threading

# Wireless and Capsense
from wireless import wifi
from murata.lbee5kl1dx import lbee5kl1dx as wifi_driver
from cypress.capsense import capsense

# AWS
from aws.iot import iot, default_credentials

# Sensor
from bosch.bme280 import bme280

# Ethereum
import eth
import config


# Configure serial and leds
streams.serial()
config.led_init()

```

Complete code at: https://github.com/zerynth/demo-ew19-firmware/blob/master/main.py

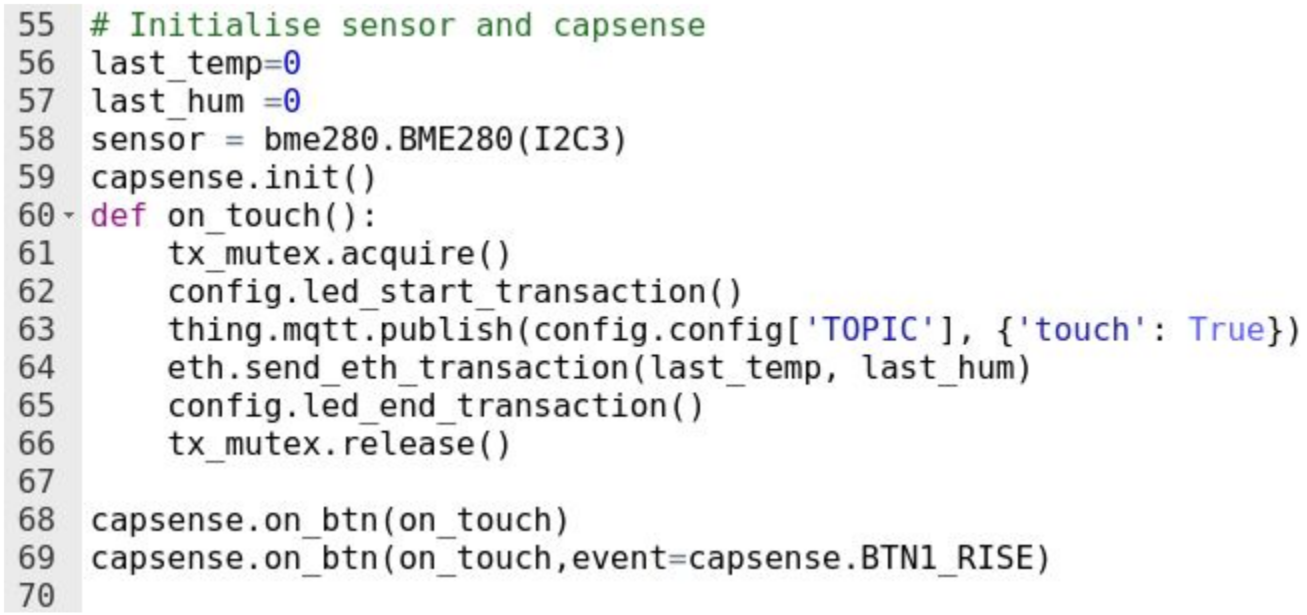
Full tutorial at: https://www.zerynth.com/blog/zerynth-and-cypress-tutorial-python-on-psoc-6-microcontrollers-for-iot-and-blockchain-applications/

Hello Blockchain: Ethereum IoT demo



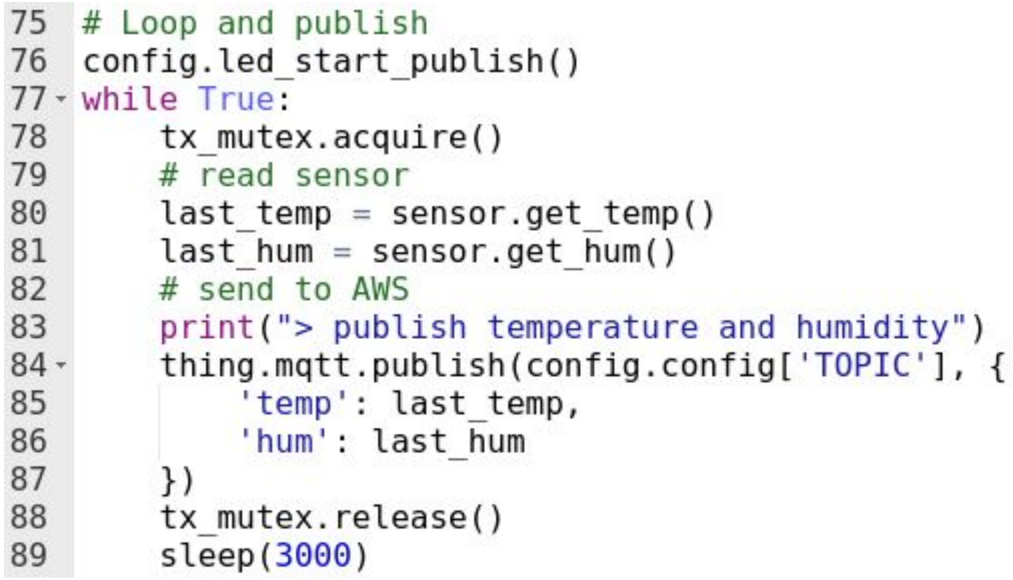
```python
# Try linking to Wifi
wifi_driver.init("US")
for _ in range(3):
    try:
        print("> Establishing Link...")
        wifi.link(config.config['SSID'],wifi.WIFI_WPA2,config.config['PSW'])
        break
    except Exception as e:
        print("> ooops, something wrong while linking :(")
else:
    mcu.reset()
print("> linked!")

# Connect to AWS IoT Core
tx_mutex = threading.Lock()
endpoint, thingname, clicert, pkey = default_credentials.load()
# derive unique id from mcu uid
mqtt_id = ''.join(['%02x' % byte for byte in mcu.uid()])
thing = iot.Thing(endpoint, mqtt_id, clicert, pkey, thingname=thingname)

print("> connecting to mqtt broker...")
thing.mqtt.connect()
print("> connected")
thing.mqtt.loop()
```

Hello Blockchain: Ethereum IoT demo



```python
# Initialise sensor and capsense
last_temp=0
last_hum =0
sensor = bme280.BME280(I2C3)
capsense.init()
def on_touch():
    tx_mutex.acquire()
    config.led_start_transaction()
    thing.mqtt.publish(config.config['TOPIC'], {'touch': True})
    eth.send_eth_transaction(last_temp, last_hum)
    config.led_end_transaction()
    tx_mutex.release()

capsense.on_btn(on_touch)
capsense.on_btn(on_touch,event=capsense.BTN1_RISE)

```

Hello Blockchain: Ethereum IoT demo



```python
# Loop and publish
config.led_start_publish()
while True:
    tx_mutex.acquire()
    # read sensor
    last_temp = sensor.get_temp()
    last_hum = sensor.get_hum()
    # send to AWS
    print("> publish temperature and humidity")
    thing.mqtt.publish(config.config['TOPIC'], {
        'temp': last_temp,
        'hum': last_hum
    })
    tx_mutex.release()
    sleep(3000)
```

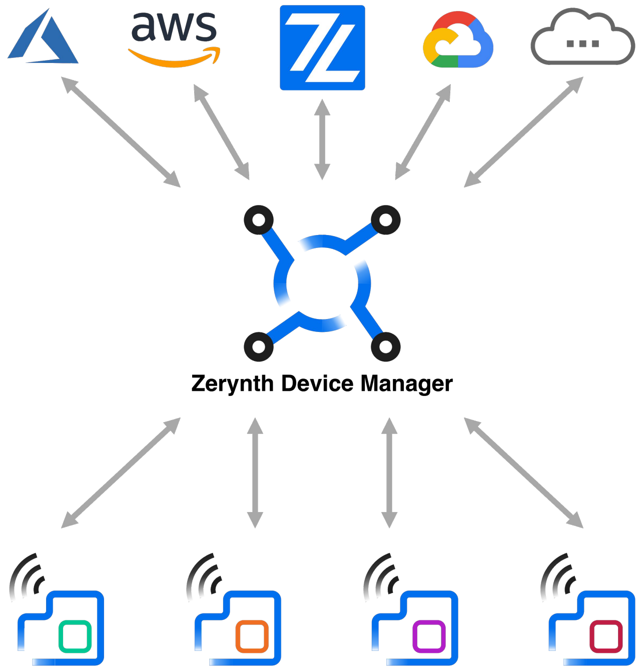
Going To Production

Using the Zerynth Device Manager to simplify the device management

The Zerynth Device Manager (ZDM) is a device and data management service that makes it easy to securely register, organize, monitor, and remotely manage IoT devices at scale.

Main Features

- **Industrial-grade security**
- **Device independent**
- **Data collection**
- **Events collection**
- **Device control**
- **Firmware Over The Air (FOTA) updates**



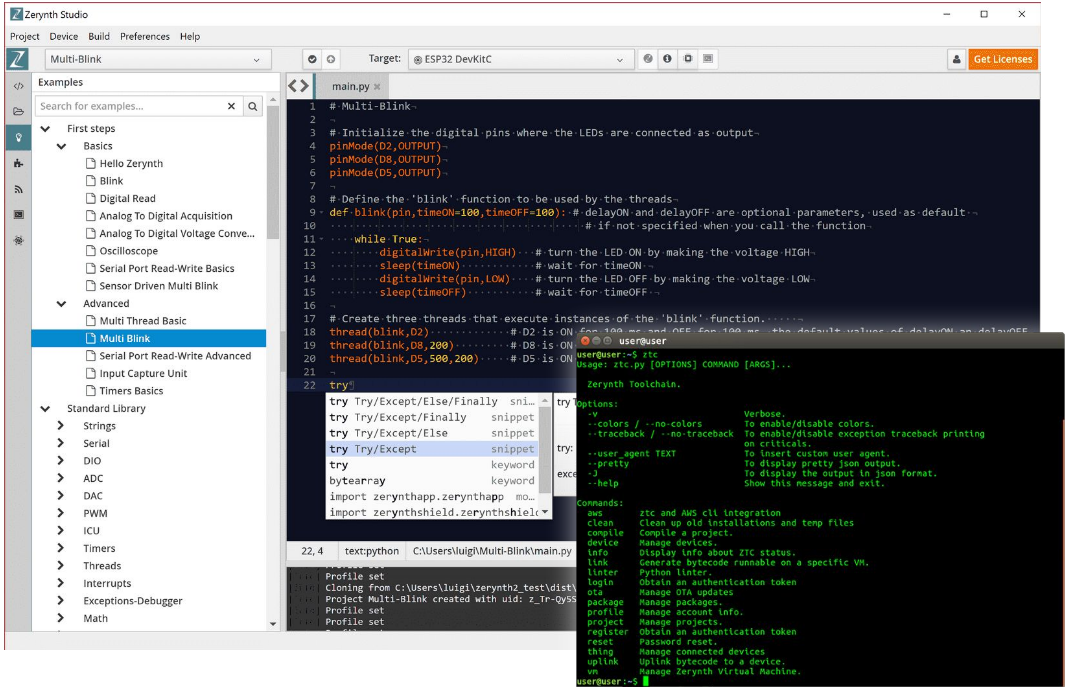
Get Started Now

Download the free Zerynth SDK and start developing with Zerynth OS on Cypress PSoC 6

The **Zerynth SDK** is the gateway to our platform and includes:

- **The Zerynth Toolchain** – a command-line interface that integrates all the essential functions for the development with Zerynth OS and the management of the Zerynth Device Manager cloud service.
- **The Zerynth Studio** – an advanced IDE for the Zerynth Toolchain. It includes development and debugging tools and numerous code examples.

https://www.zerynth.com/zsdk/



```python
# Multi-Blink

# Initialize the digital pins where the LEDs are connected as output
pinMode(D2,OUTPUT)
pinMode(D8,OUTPUT)
pinMode(D5,OUTPUT)

# Define the 'blink' function to be used by the threads
def blink(pin,timeON=100,timeOFF=100): # delayON and delayOFF are optional parameters, used as default
                                        # if not specified when you call the function
    while True:
        digitalWrite(pin,HIGH)   # turn the LED ON by making the voltage HIGH
        sleep(timeON)            # wait for timeON
        digitalWrite(pin,LOW)    # turn the LED OFF by making the voltage LOW
        sleep(timeOFF)           # wait for timeOFF

# Create three threads that execute instances of the 'blink' function.
thread(blink,D2)             # D2 is ON for 100 ms and OFF for 100 ms, the default values of delayON an delayOFF
thread(blink,D8,200)         # D8 is ON
thread(blink,D5,500,200)     # D5 is ON

try
```

```
Profile set
Cloning from C:\Users\luigi\zerynth2_test\dist\
Project Multi-Blink created with uid: z_Tr-Qy5S
Profile set
Profile set
```

```
user@user:~$ ztc
Usage: ztc.py [OPTIONS] COMMAND [ARGS]...

  Zerynth Toolchain.

Options:
  -v                          Verbose.
  --colors / --no-colors      To enable/disable colors.
  --traceback / --no-traceback  To enable/disable exception traceback printing
                              on criticals.
  --user_agent TEXT           To insert custom user agent.
  --pretty                    To display pretty json output.
  -J                          To display the output in json format.
  --help                      Show this message and exit.

Commands:
  aws       ztc and AWS cli integration
  clean     Clean up old installations and temp files
  compile   Compile a project.
  device    Manage devices.
  info      Display info about ZTC status.
  link      Generate bytecode runnable on a specific VM.
  linter    Python linter.
  login     Obtain an authentication token
  ota       Manage OTA updates
  package   Manage packages.
  profile   Manage account info.
  project   Manage projects.
  register  Obtain an authentication token
  reset     Password reset.
  thing     Manage connected devices
  uplink    Uplink bytecode to a device.
  vm        Manage Zerynth Virtual Machine.
user@user:~$
```

Zerynth®

Thank you!

www.zerynth.com