



**FACULTAD
DE INGENIERIA**

Universidad de Buenos Aires

Presentación ISO II

Maestría en Sistemas Embebidos

Implementación de Driver de sensor HTU21D y desarrollo de daemon en Python para suba de datos automática a plataforma IoT de Adafruit

Autor: Esp. Ing. Marcelo Daniel Pistarelli

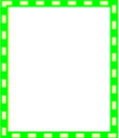
Trabajo realizado

- Recompilación de kernel con driver desarrollado en IMD
- Generación de root filesystem con buildroot
 - ✓ Cliente DHCP
 - ✓ Cliente NTP
 - ✓ Python 3
 - ✓ Request
 - ✓ SSL
 - ✓ Servidor SSH dropbear

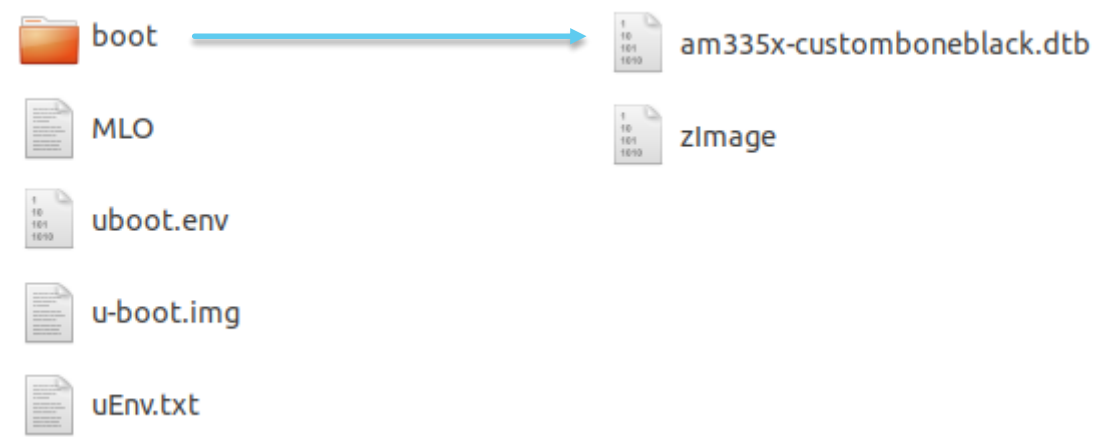
Trabajo realizado

- Recompilación de u-boot v2019.01
 - ✓ Sin parches
 - ✓ Configurado para soportar saveenv en partición fat
 - ✓ Generación de archivo uEnv.txt
- Particionado de tarjeta SD
 - ✓ Partición FAT boot de 256MB
 - ✓ Partición EXT4 para alojar root filesystem

Contenido de tarjeta SD

|  | | <div>/dev/mmcblk0p2 3.39 GiB</div> | | | | | |
|---|---------------------|--|----------|------------|------------|------------|-----------|
| Partición | Sistema de archivos | Punto de montaje | Etiqueta | Tamaño | Usado | Libre | Opciones |
| /dev/mmcblk0p1 | fat16 | /media/marcel... | | 256.00 MiB | 5.07 MiB | 250.93 MiB | boot, lba |
| /dev/mmcblk0p2 | ext4 | /media/marcel... | datos | 3.39 GiB | 217.79 MiB | 3.17 GiB | |

- /dev/mmcblk0p1



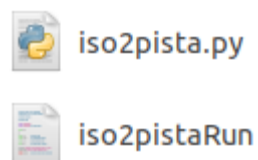
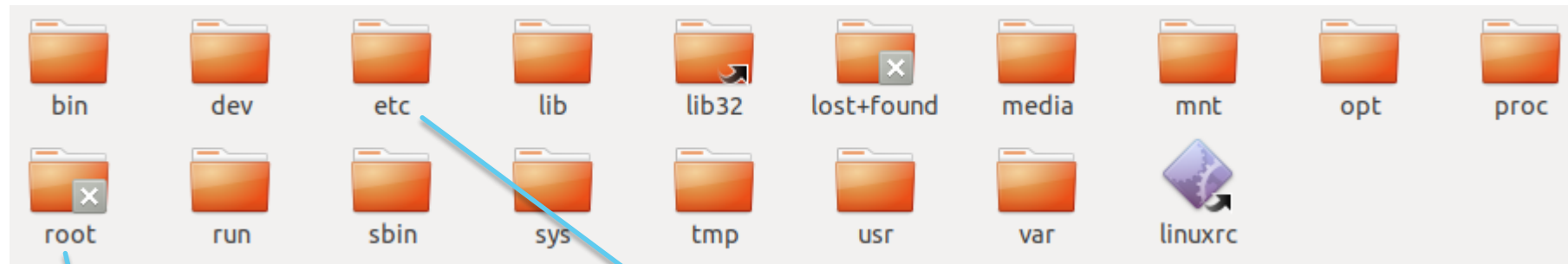
Contenido de tarjeta SD

- uEnv.txt

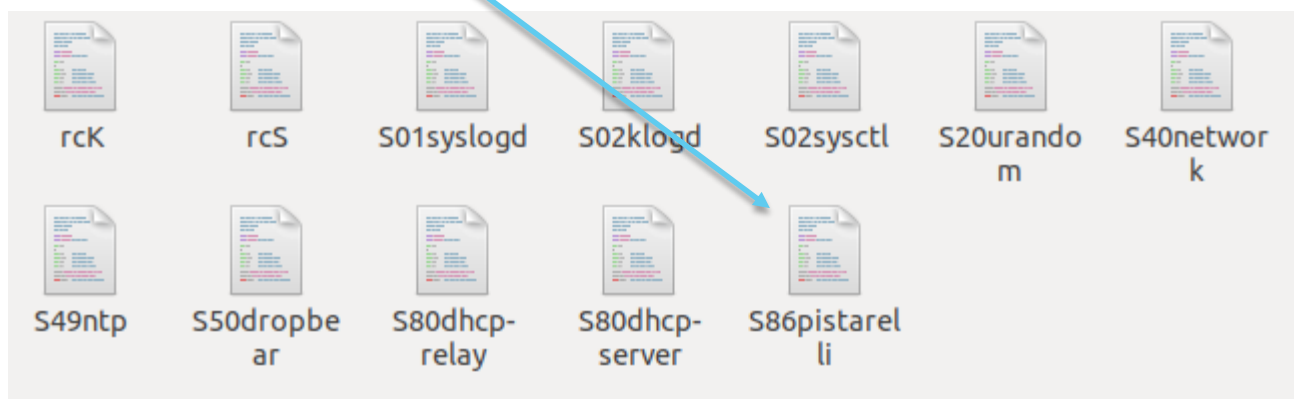
```
loadzimage=fatload mmc 0:1 0x82000000 boot/zImage  
loadfdt=fatload mmc 0:1 0x88000000 boot/am335x-customboneblack.dtb  
my_bootargs=setenv bootargs console=tty00,115200n8 root=/dev/mmcblk0p2  
uenvcmd=run loadzimage; run loadfdt; run my_bootargs; bootz 0x82000000 - 0x88000000
```

Contenido de tarjeta SD

- `/dev/mmcblk0p2`



`/etc/init.d/`



Formato de nombre: `S(0-99)nombre del script`

Contenido de tarjeta SD

- S86pistarelli

```
#!/bin/sh
```

```
DIR=/root
DAEMON=$DIR/iso2pistaRun
DAEMON_NAME=iso2pista
```

```
do_start () {
    printf "Starting $DAEMON_NAME daemon"
    start-stop-daemon -S -q -x $DAEMON -- -g
    [ $? = 0 ] && echo "OK" || echo "FAIL"
}
do_stop () {
    printf "Stopping $DAEMON_NAME daemon"
    start-stop-daemon -K -q -n $DAEMON_NAME
    [ $? = 0 ] && echo "OK" || echo "FAIL"
}
```

```
case "$1" in
```

```
    start|stop)
        do_${1}
        ;;
```

```
    restart|reload|force-reload)
        do_stop
        do_start
        ;;
```

```
    *)
        echo "Usage: /etc/init.d/$DAEMON_NAME {start|stop|restart}"
        exit 1
        ;;
```

```
esac
exit 0
```

- iso2pistaRun

```
#!/bin/sh
```

```
python /root/iso2pista.py
```

Contenido de tarjeta SD

- Iso2pista.py

```
import os
import signal
import requests
import time
import sys

data = bytearray(4)

# Datos de acceso a Adafruit
API_DATA_TEMP = "https://io.adafruit.com/api/v2/pista86/feeds/temp/data"
API_DATA_HUM = "https://io.adafruit.com/api/v2/pista86/feeds/hum/data"
API_KEY = "a2e11cf933ba4a548d10684762dd4393"
header = {'X-AIO-Key': API_KEY}

def receiveSignal(signalNumber, frame):
    os.close(fd)
    print("Device cerrado")
    sys.exit()

def sendHum(humVal):
    datos = {'value': '{0:.2f}'.format(humVal)}
    r = requests.post(url = API_DATA_HUM, data=datos, headers=header)
    # debug
    print(r.text)
    print("Humedad enviada a servidor:%0.2f"%humVal)

def sendTemp(tempVal):
    datos = {'value': '{0:.2f}'.format(tempVal)}
    r = requests.post(url = API_DATA_TEMP, data=datos, headers=header)
    # debug
    print(r.text)
    print("Temperatura enviada a servidor:%0.2f"%tempVal)
```


Contenido de tarjeta SD

- Iso2pista.py

```
# Registrar handler de señal
signal.signal(signal.SIGINT, receiveSignal)

# Apertura de device
fd = os.open("/dev/i2c-pistahtu21d", os.O_RDWR)

if (fd != 0):
    print("Device abierto")
    medirTyH = 1
    resol = 14
    os.write(fd, resol.to_bytes(1, byteorder='big', signed=False))

while (medirTyH):
    os.readv(fd, [data])

    if len(data) == 4 :
        humedad = ((data[1] << 8) + data[0]) & 0xFFFFC
        temperatura = ((data[3] << 8) + data[2]) & 0xFFFFC

        humedadFloat = ((humedad * 125.0) / 65536) - 6
        temperaturaFloat = ((temperatura * 175.72) / 65536) - 46.85

        sendHum(humedadFloat)
        time.sleep(10)
        sendTemp(temperaturaFloat)

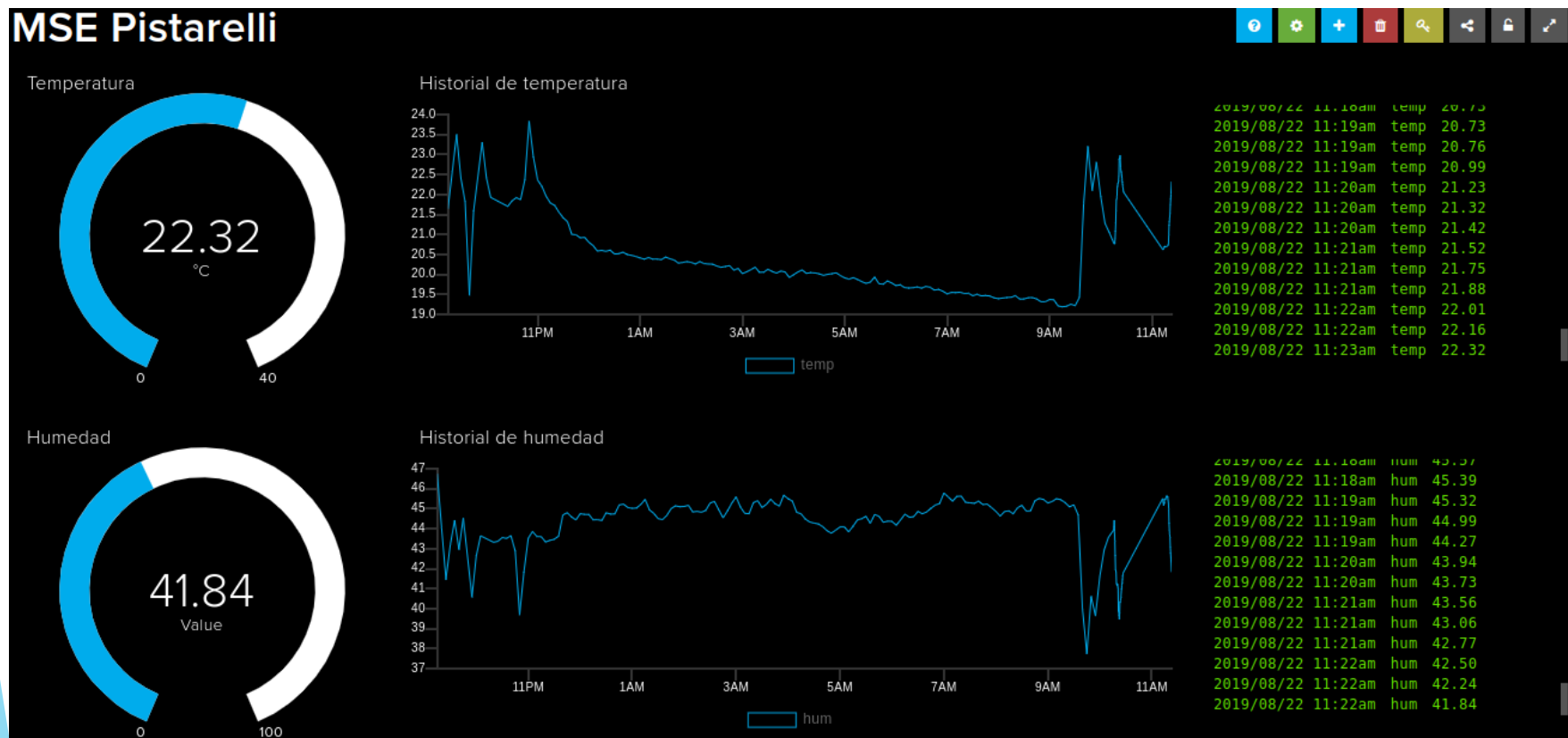
    time.sleep(10)
```

Plataforma IoT

<https://io.adafruit.com/pista86/dashboards/mse-pistarelli>

Se debe tener cuenta en Adafruit para visualizar los datos !

MSE Pistarelli



The background features abstract, overlapping geometric shapes in various shades of blue, ranging from light sky blue to deep navy blue. These shapes are primarily located on the right side of the image, creating a modern, dynamic feel. The word "Prueba" is centered on the left side of the image.

Prueba