

IoT Chile
Meetup

Bienvenidos!

Agradecimientos a:

HUB
providencia

Agradecimientos a:

DuocUC[®]



Agradecimientos a:



ALLIED
ELECTRONICS & AUTOMATION

Quienes somos?



**Felipe
Molina**

Lead SW dev

Wisely, una empresa de desarrollo de tecnología a medida, con foco en soluciones de Internet de las Cosas Industriales (IIoT).

Por mi parte, Ingeniero eléctrico de la USACH, desarrollador de software (Akamai, US Energy Engineers), soluciones IoT (P&P, Wisely)



Síguenos

meetup

meetup.com/loT-Chile-Meetup



facebook.com/wisely.chile



twitter.com/wisely_cl



linkedin.com/company/wiselycl





Como partir y no
morir en el intento

Taller de IoT Aplicado

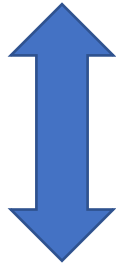
IoT Chile



Meetup

- I – Conectándose a raspberry PI (SSH + terminal).
- II - Electrónica aplicada (LED, pulsador, GPIO y Python).
- III – Electrónica más aplicada (Python + Sensor temperatura + 1W).
- IV – Enviando datos a la web (CURL + Python).
- V – Enviando datos a la web, bien (Request + JSON).
- VI - Ahora, todos juntos (Portal web con datos del equipo).

Conectándose a RPI



1. Windows 10 actualizado o Linux, de no ser así, favor descargar putty (Wifi)
2. Compartir conexión a internet (share internet connection)
3. Encontrar la IP de la raspberry:
 - A. Windows:
 - I. Encontrar IP Propia (ipconfig)
 - II. Ping a dirección broadcast (192.X.X.255)
 - III. Arp -a y buscar la ip distinta a la nuestra
 - IV. Ssh pi@<ip encontrada>
 - V. Pass: Raspberry
 - B. Linux:
 - I. Nmap -sP <ippropia>
 - II. Ssh a nueva ip

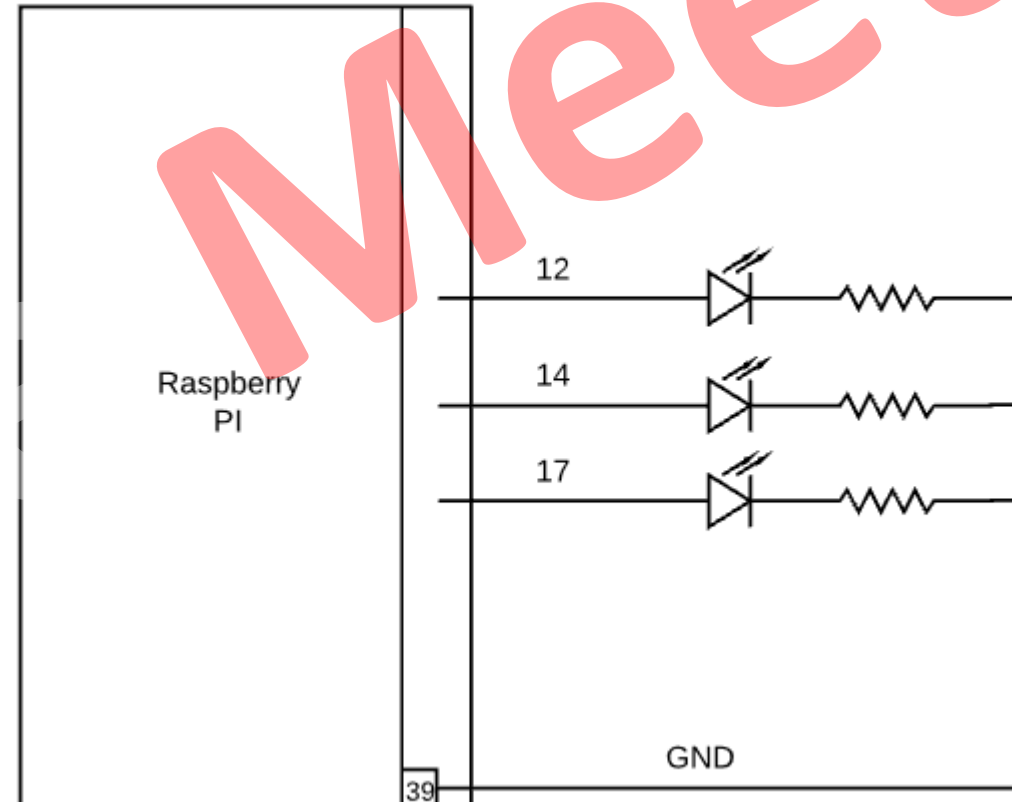
Raspberry Pi Pinout

3v3 Power	1	2	5v Power
BCM 2 (WiringPi 8)	3	4	5v Power
BCM 3 (WiringPi 9)	5	6	Ground
BCM 4 (WiringPi 7)	7	8	BCM 14 (WiringPi 15)
Ground	9	10	BCM 15 (WiringPi 16)
BCM 17 (WiringPi 0)	11	12	BCM 18 (WiringPi 1)
BCM 27 (WiringPi 2)	13	14	Ground
BCM 22 (WiringPi 3)	15	16	BCM 23 (WiringPi 4)
3v3 Power	17	18	BCM 24 (WiringPi 5)
BCM 10 (WiringPi 12)	19	20	Ground
BCM 9 (WiringPi 13)	21	22	BCM 25 (WiringPi 6)
BCM 11 (WiringPi 14)	23	24	BCM 8 (WiringPi 10)
Ground	25	26	BCM 7 (WiringPi 11)
BCM 0 (WiringPi 30)	27	28	BCM 1 (WiringPi 31)
BCM 5 (WiringPi 21)	29	30	Ground
BCM 6 (WiringPi 22)	31	32	BCM 12 (WiringPi 26)
BCM 13 (WiringPi 23)	33	34	Ground
BCM 19 (WiringPi 24)	35	36	BCM 16 (WiringPi 27)
BCM 26 (WiringPi 25)	37	38	BCM 20 (WiringPi 28)
Ground	39	40	BCM 21 (WiringPi 29)

1. Conectadando componentes en la Protoboard.
2. LEDs.
3. Pulsador.
4. Conectando a RPI.
5. Python
6. ¡Encendiendo una LED!

Electronica Aplicada I

Raspberry Pi Pinout			
3v3 Power	1	2	5v Power
BCM 2 (WiringPi 8)	3	4	5v Power
BCM 3 (WiringPi 9)	5	6	Ground
BCM 4 (WiringPi 7)	7	8	BCM 14 (WiringPi 15)
Ground	9	10	BCM 15 (WiringPi 16)
BCM 17 (WiringPi 0)	11	12	BCM 18 (WiringPi 1)
BCM 27 (WiringPi 2)	13	14	Ground
BCM 22 (WiringPi 3)	15	16	BCM 23 (WiringPi 4)
3v3 Power	17	18	BCM 24 (WiringPi 5)
BCM 10 (WiringPi 12)	19	20	Ground
BCM 9 (WiringPi 13)	21	22	BCM 25 (WiringPi 6)
BCM 11 (WiringPi 14)	23	24	BCM 8 (WiringPi 10)
Ground	25	26	BCM 7 (WiringPi 11)
BCM 0 (WiringPi 30)	27	28	BCM 1 (WiringPi 31)
BCM 5 (WiringPi 21)	29	30	Ground
BCM 6 (WiringPi 22)	31	32	BCM 12 (WiringPi 26)
BCM 13 (WiringPi 23)	33	34	Ground
BCM 19 (WiringPi 24)	35	36	BCM 16 (WiringPi 27)
BCM 26 (WiringPi 25)	37	38	BCM 20 (WiringPi 28)
Ground	39	40	BCM 21 (WiringPi 29)



Raspberry Pi Pinout

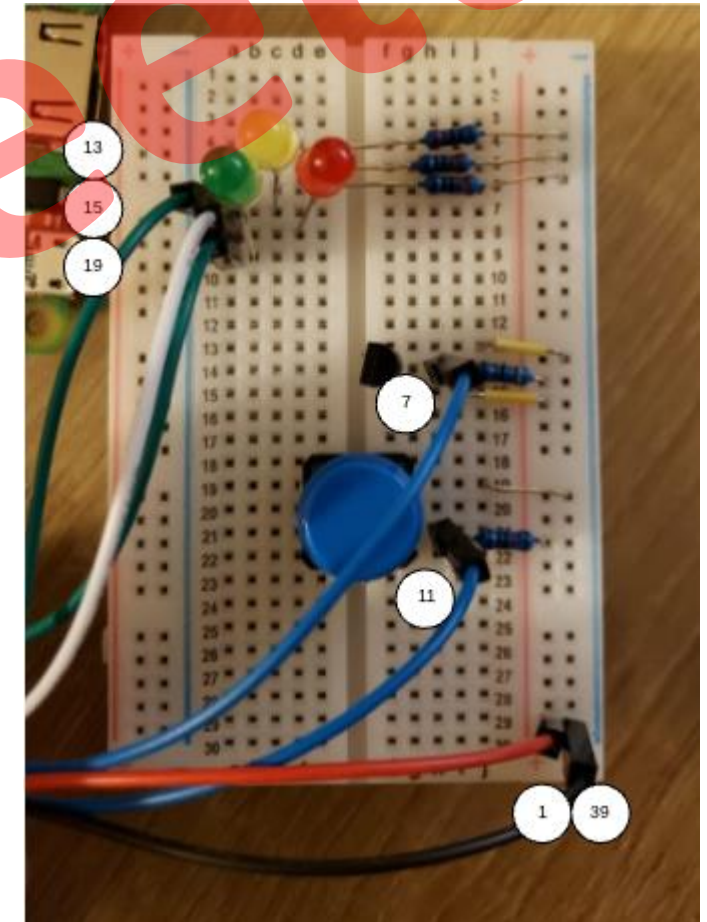
3v3 Power	1	2	5v Power
BCM 2 (WiringPi 8)	3	4	5v Power
BCM 3 (WiringPi 9)	5	6	Ground
BCM 4 (WiringPi 7)	7	8	BCM 14 (WiringPi 15)
Ground	9	10	BCM 15 (WiringPi 16)
BCM 17 (WiringPi 0)	11	12	BCM 18 (WiringPi 1)
BCM 27 (WiringPi 2)	13	14	Ground
BCM 22 (WiringPi 3)	15	16	BCM 23 (WiringPi 4)
3v3 Power	17	18	BCM 24 (WiringPi 5)
BCM 10 (WiringPi 12)	19	20	Ground
BCM 9 (WiringPi 13)	21	22	BCM 25 (WiringPi 6)
BCM 11 (WiringPi 14)	23	24	BCM 8 (WiringPi 10)
Ground	25	26	BCM 7 (WiringPi 11)
BCM 0 (WiringPi 30)	27	28	BCM 1 (WiringPi 31)
BCM 5 (WiringPi 21)	29	30	Ground
BCM 6 (WiringPi 22)	31	32	BCM 12 (WiringPi 26)
BCM 13 (WiringPi 23)	33	34	Ground
BCM 19 (WiringPi 24)	35	36	BCM 16 (WiringPi 27)
BCM 26 (WiringPi 25)	37	38	BCM 20 (WiringPi 28)
Ground	39	40	BCM 21 (WiringPi 29)

Ahora, desafio I:

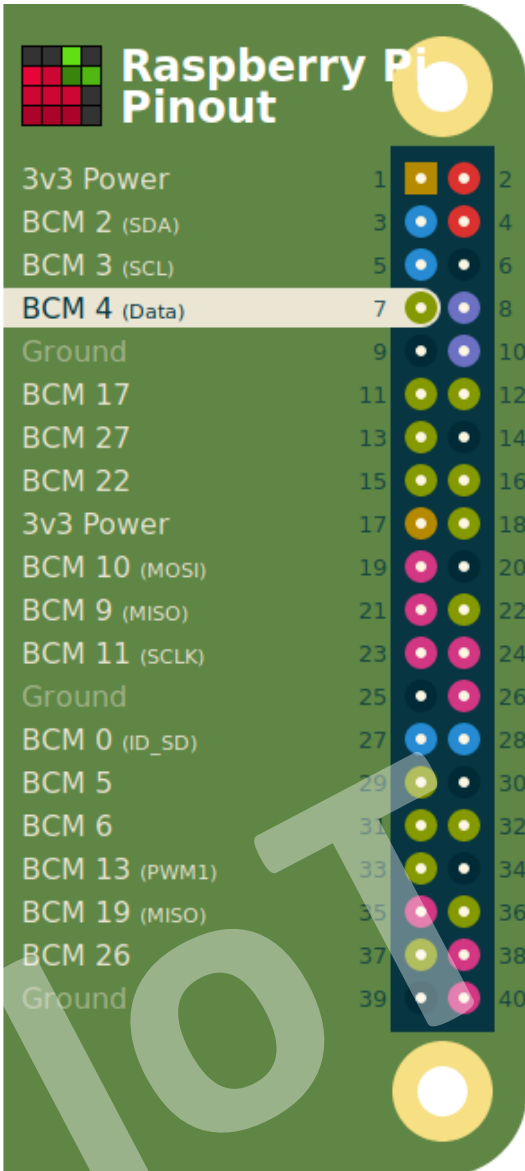
- Conectar dos leds adicionales y hacer una secuencia de luces!
- combinar leds y botones (al presionar un boton, activar una led en especifico).

Electronica Aplicada I

Raspberry Pi Pinout			
3v3 Power	1	2	5v Power
BCM 2 (WiringPi 8)	3	4	5v Power
BCM 3 (WiringPi 9)	5	6	Ground
BCM 4 (WiringPi 7)	7	8	BCM 14 (WiringPi 15)
Ground	9	10	BCM 15 (WiringPi 16)
BCM 17 (WiringPi 0)	11	12	BCM 18 (WiringPi 1)
BCM 27 (WiringPi 2)	13	14	Ground
BCM 22 (WiringPi 3)	15	16	BCM 23 (WiringPi 4)
3v3 Power	17	18	BCM 24 (WiringPi 5)
BCM 10 (WiringPi 12)	19	20	Ground
BCM 9 (WiringPi 13)	21	22	BCM 25 (WiringPi 6)
BCM 11 (WiringPi 14)	23	24	BCM 8 (WiringPi 10)
Ground	25	26	BCM 7 (WiringPi 11)
BCM 0 (WiringPi 30)	27	28	BCM 1 (WiringPi 31)
BCM 5 (WiringPi 21)	29	30	Ground
BCM 6 (WiringPi 22)	31	32	BCM 12 (WiringPi 26)
BCM 13 (WiringPi 23)	33	34	Ground
BCM 19 (WiringPi 24)	35	36	BCM 16 (WiringPi 27)
BCM 26 (WiringPi 25)	37	38	BCM 20 (WiringPi 28)
Ground	39	40	BCM 21 (WiringPi 29)



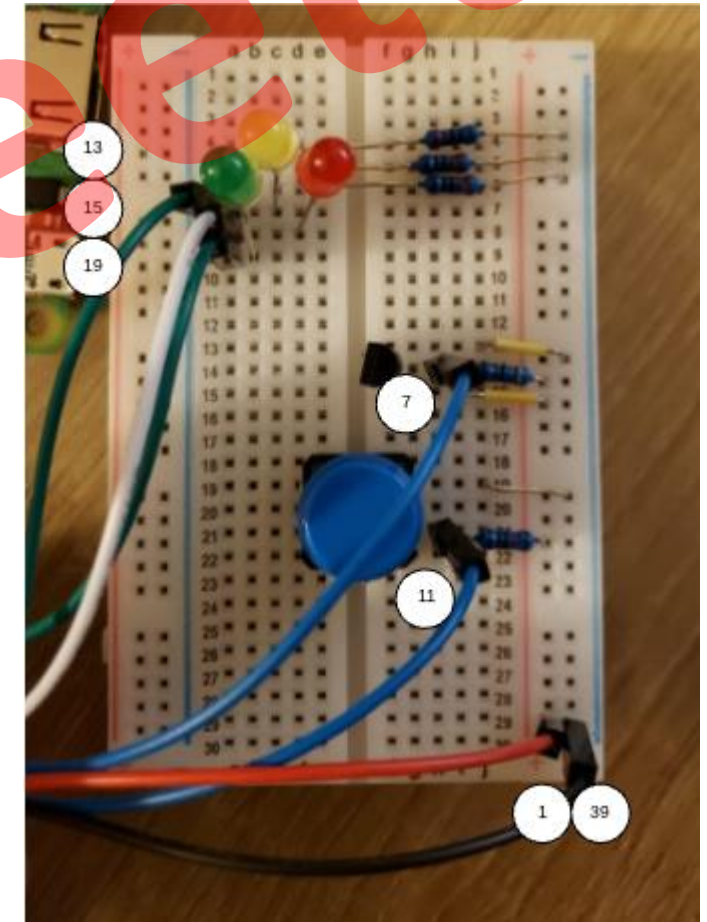
Electronica MAS Aplicada II



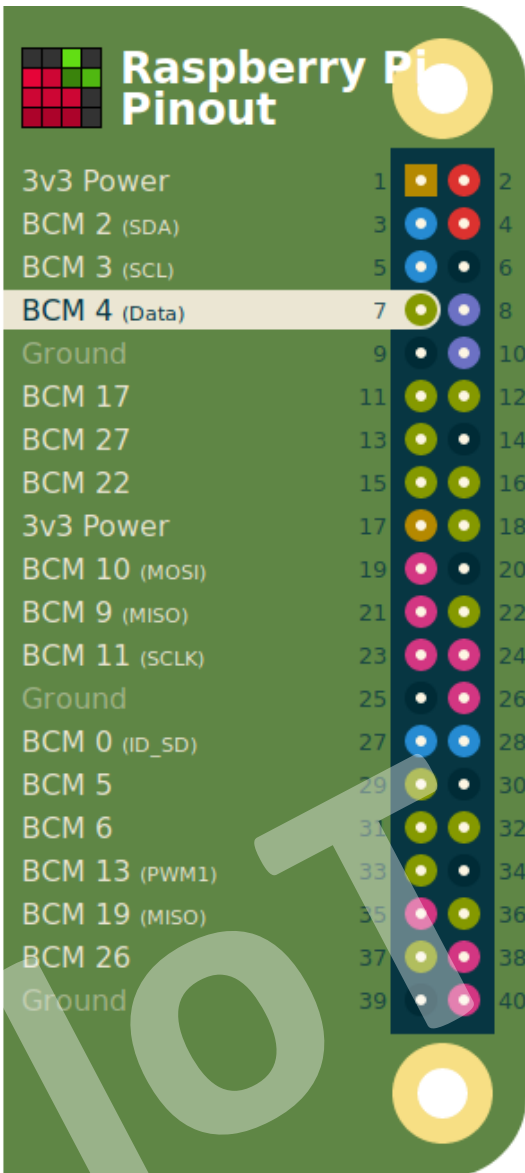
- Sensor de temperatura (one wire - 1W).
- Circuito necesario.
- Configuración en RPI.
- Lectura temperatura.

Electronica Aplicada I

Raspberry Pi Pinout			
3v3 Power	1	2	5v Power
BCM 2 (WiringPi 8)	3	4	5v Power
BCM 3 (WiringPi 9)	5	6	Ground
BCM 4 (WiringPi 7)	7	8	BCM 14 (WiringPi 15)
Ground	9	10	BCM 15 (WiringPi 16)
BCM 17 (WiringPi 0)	11	12	BCM 18 (WiringPi 1)
BCM 27 (WiringPi 2)	13	14	Ground
BCM 22 (WiringPi 3)	15	16	BCM 23 (WiringPi 4)
3v3 Power	17	18	BCM 24 (WiringPi 5)
BCM 10 (WiringPi 12)	19	20	Ground
BCM 9 (WiringPi 13)	21	22	BCM 25 (WiringPi 6)
BCM 11 (WiringPi 14)	23	24	BCM 8 (WiringPi 10)
Ground	25	26	BCM 7 (WiringPi 11)
BCM 0 (WiringPi 30)	27	28	BCM 1 (WiringPi 31)
BCM 5 (WiringPi 21)	29	30	Ground
BCM 6 (WiringPi 22)	31	32	BCM 12 (WiringPi 26)
BCM 13 (WiringPi 23)	33	34	Ground
BCM 19 (WiringPi 24)	35	36	BCM 16 (WiringPi 27)
BCM 26 (WiringPi 25)	37	38	BCM 20 (WiringPi 28)
Ground	39	40	BCM 21 (WiringPi 29)



Electronica MAS Aplicada II



Desafio II: Alarma de temperatura

- Revisar la temperatura cada un segundo, si el valor supera los 20°C, prender todas las luces en secuencia.

- Si la temperatura baja o sube, prender una luz.

- Portal del taller (¡Preguntar por IP!).
- Como se envían datos.
- Utilizando Requests para el envío de datos



Envió de datos via HTTP

```
1 import requests
2
3 payload = {
4     'marco': 'polo!'
5 }
6
7 res = requests.post('http://localhost:5000/echo', json=payload)
8 if res.ok:
9     print(res.json())
```

Cliente:

```
(venv) kachu@Walkyrie:~/Documents/Projects/taller_iot$ python envio_datos.py
{'status': 'Mensaje exitoso!'}
(venv) kachu@Walkyrie:~/Documents/Projects/taller_iot$
```

Servidor:

```
{'marco': 'polo!'}
127.0.0.1 - - [03/Jul/2019 22:36:15] "POST /echo HTTP/1.1" 200
```

- Código del servidor

```
@app.route('/echo', methods=['POST'])
def receive_data_test():
    content = request.json
    if content:
        print(content)
        return jsonify({"status": "Mensaje exitoso!"})
    return jsonify({"status": "no hay datos :<"})

@app.route('/nuevo_usuario', methods=['POST'])
def receive_data():
    content = request.json
    try:
        if content['id_grupo'] in user_data:
            return jsonify({"status": "Usuario ya existe!"})

        user_data[content['id_grupo']] = {
            'id_grupo': content['id_grupo'],
            'nombre_grupo': content['nombre_grupo'],
            'integrantes': content['integrantes']
        }
        return jsonify({"status": "Usuario creado!"})
    except KeyError as e:
        return jsonify({"status": "Error en campo!", "campo con error": e.__str__()})
    return jsonify({"status": "Usuario creado!"})
```

- Payload esperada

```
1
2 Info_del_grupo = {
3     'id_grupo': '',
4     'nombre_grupo': '',
5     'integrantes': ''
6 }
```

Desafío Final: Utilizar servidor propio para el envío de datos y realizar acciones

Meetup