Ejercicio

MEDICIÓN DE LA DISTANCIA Y LA LUMINOSIDAD

EJERCICIO 02 – SESIÓN 04

2020/2021

DISEÑO DE SISTEMAS OPERATIVOS

PROPÓSITO

El primer objetivo de este ejercicio consiste en medir la distancia de objetos con un sensor ultrasónico y una Raspberry Pi

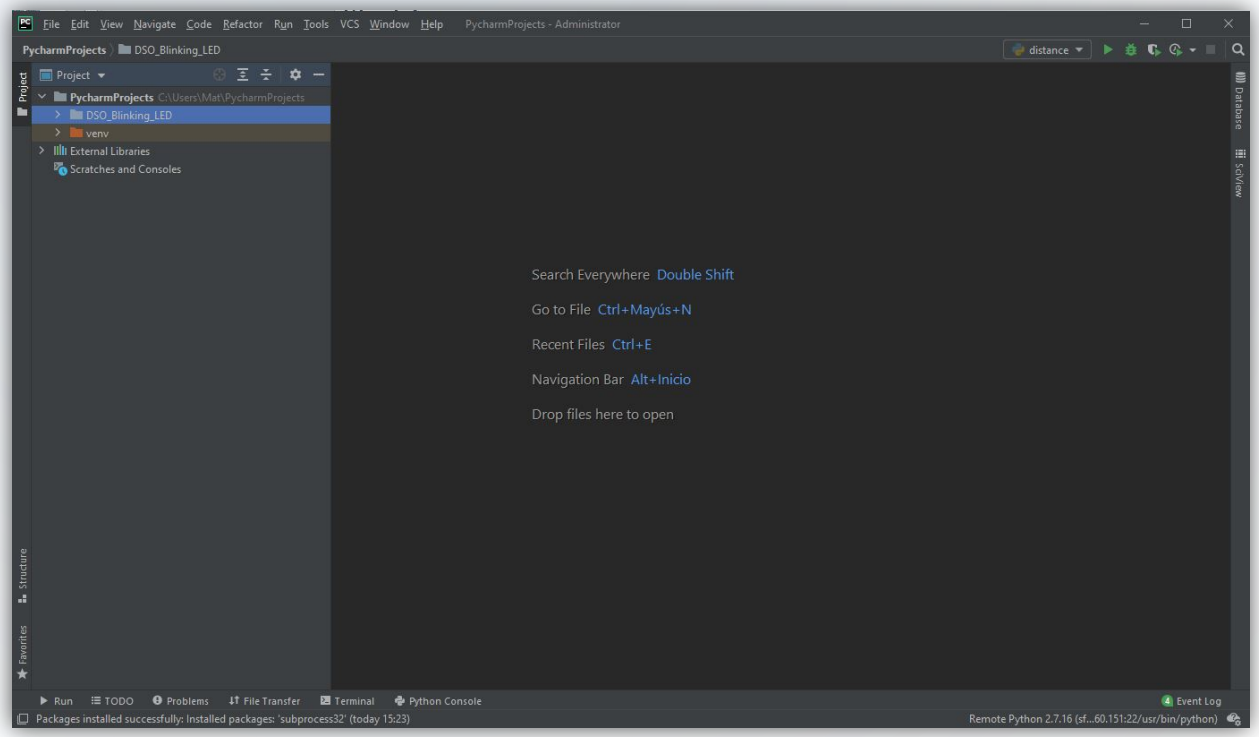
El segundo objetivo de este ejercicio consiste en medir la luz con un sensor LDR utilizando una Raspberry Pi.

La presentación introducida por el profesor deberá usarse como guía para completar los pasos propuestos.

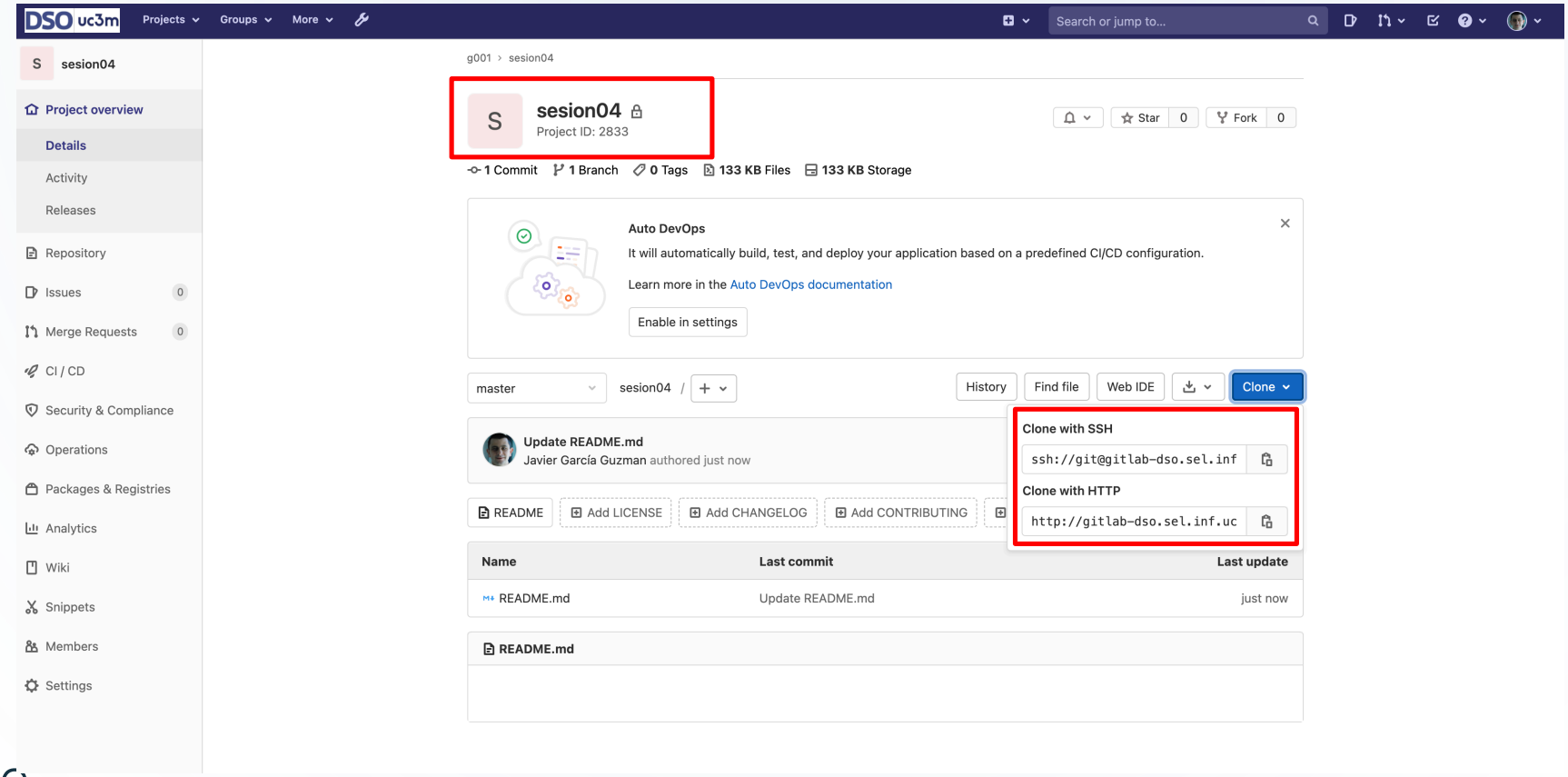
MATERIAL NECESARIO

- Raspberry Pi 3 B+
- Breadboard.
- Sensor de Ultrasonidos HC-SR04.
- Sensor LDR
- Capacitor
- Resistencia 1kΩ.
- Resistencia 2kΩ Resistor.
- Cables de conexión macho-hembra y macho-macho

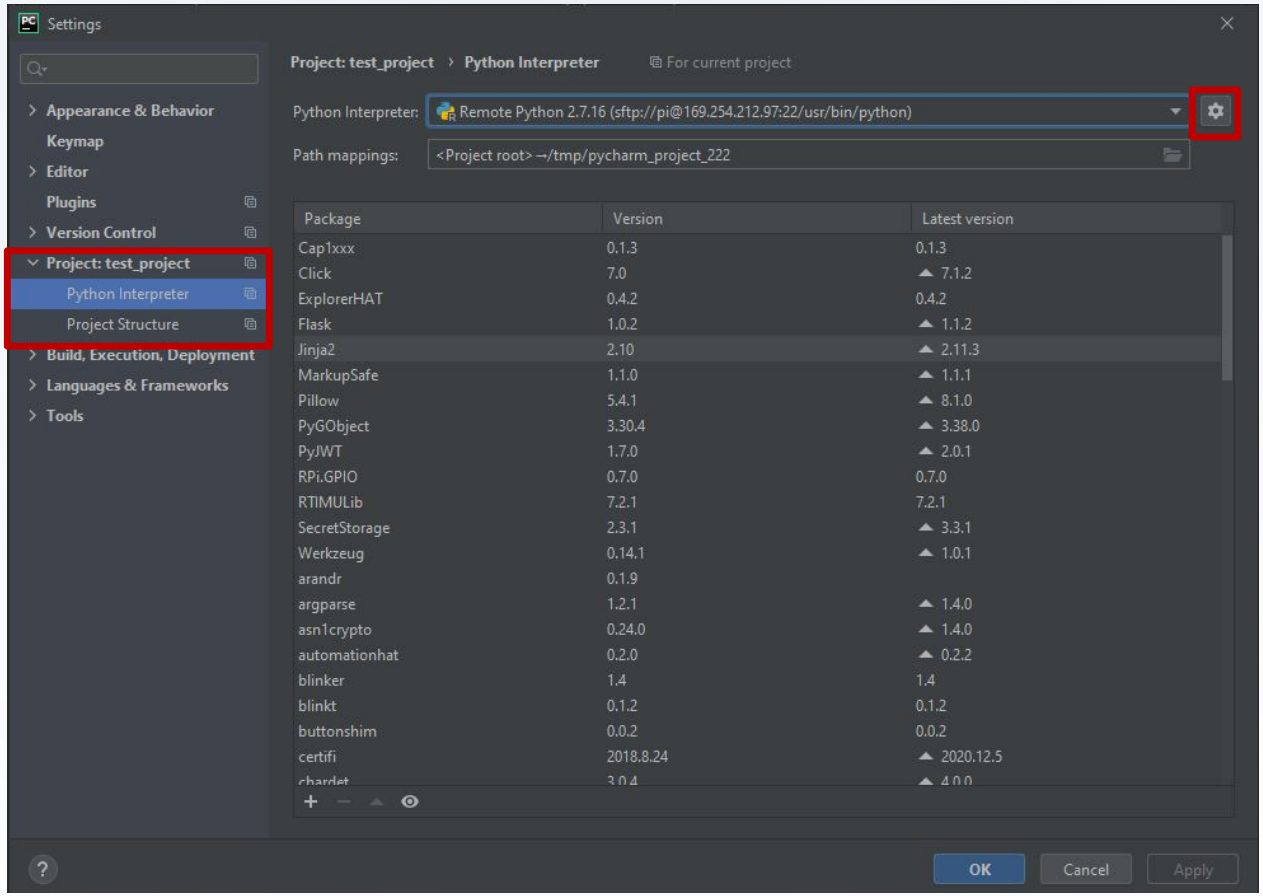
USAR ENTORNO PYCHARM PROFESIONAL



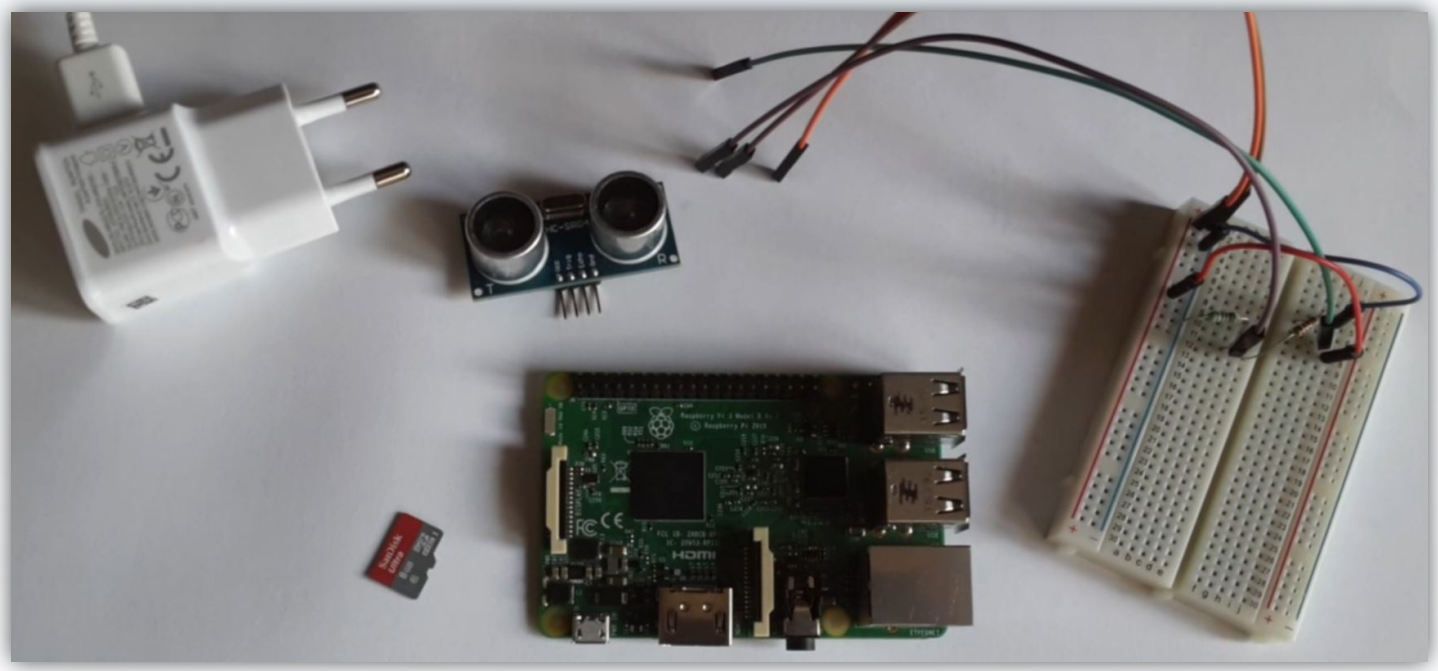
CONFIGURACIÓN DEL PROYECTO CON GITLAB



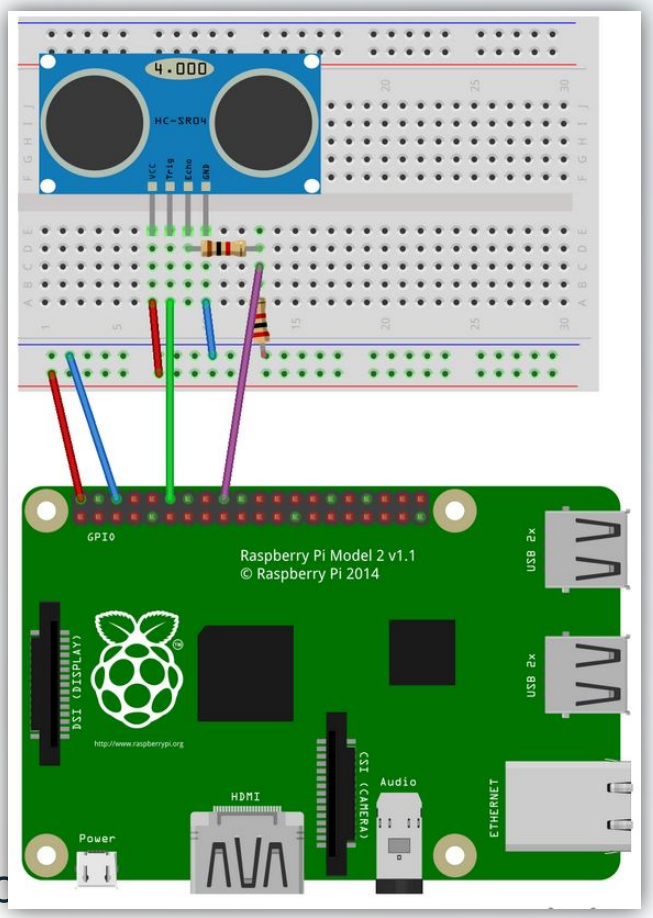
CONFIGURAR INTERPRETE REMOTO SSH



RASPBERRY PI – MEDICION DE DISTANCIA CON SENSOR DE ULTRASONIDOS HC-SR04



RASPBERRY PI – MEDICION DISTANCIA – CONEXION DE COMPONENTES



BOARD	GPIO		GPIO	BOARD
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I²C)		DC Power 5v	04
05	GPIO03 (SCL1 , I²C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)		(I²C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

1 – 5V (1)

2 – GPIO18 (6) TRIGGER

3- Resistencia – GPIO24 (9) ECHO – Resitencia – Ground (3)

4 – Ground (3)

RASPBERRY PI – MEDICIÓN DE DISTANCIA – CÓDIGO A IMPLEMENTAR

```python
# set GPIO Pins
pinTrigger = 18
pinEcho = 24

def close(signal, frame):
    print("\nTurning off ultrasonic distance detection...\n")
    GPIO.cleanup()
    sys.exit(0)

signal.signal(signal.SIGINT, close)

# set GPIO input and output channels
GPIO.setup(pinTrigger, GPIO.OUT)
GPIO.setup(pinEcho, GPIO.IN)
```

Figure a. Set up and initialize the Raspberry Pi environment.

```python
while True:
    # set Trigger to HIGH
    GPIO.output(pinTrigger, True)
    # set Trigger after 0.01ms to LOW
    time.sleep(0.00001)
    GPIO.output(pinTrigger, False)

    startTime = time.time()
    stopTime = time.time()

    # save start time
    while 0 == GPIO.input(pinEcho):
        startTime = time.time()

    # save time of arrival
    while 1 == GPIO.input(pinEcho):
        stopTime = time.time()

    # time difference between start and arrival
    TimeElapsed = stopTime - startTime
    # multiply with the sonic speed (34300 cm/s)
    # and divide by 2, because there and back
    distance = (TimeElapsed * 34300) / 2

    print("Distance: %.1f cm" % distance)
    time.sleep(1)
```

Figure b. Taking measurements to calculate the distance.

RASPBERRY PI – MEDICIÓN DE DISTANCIA – RESULTADO

RASPBERRY PI – DETECCION DE LUZ CON LDR

Figure a). LDR Sensor.

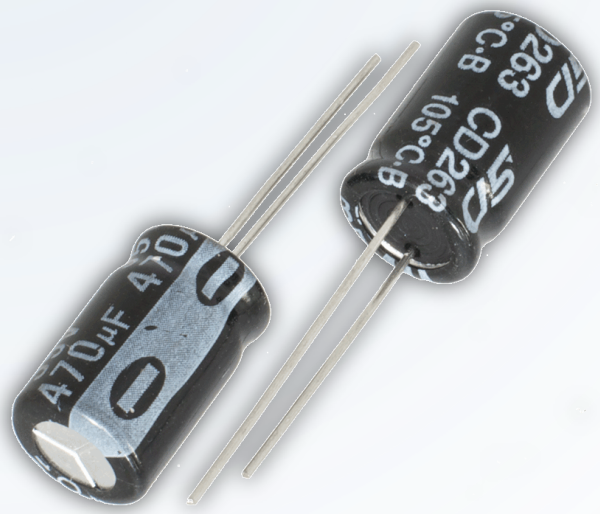
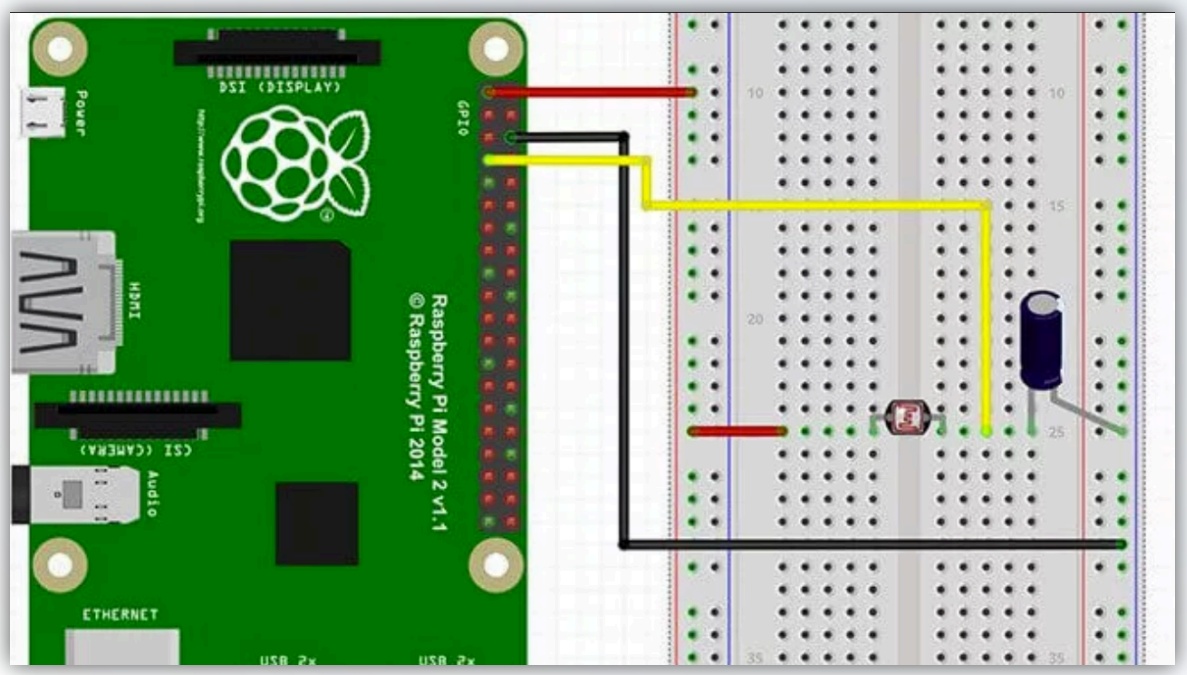


Figure b). Capacitor.

RASPBERRY PI – DETECCION DE LUZ CON LDR – CONEXION DE COMPONENTES



BOARD	GPIO		GPIO	BOARD
01	3.3v DC Power		DC Power 5v	02
03	GPIO02 (SDA1 , I²C)		DC Power 5v	04
05	GPIO03 (SCL1 , I²C)		Ground	06
07	GPIO04 (GPIO_GCLK)		(TXD0) GPIO14	08
09	Ground		(RXD0) GPIO15	10
11	GPIO17 (GPIO_GEN0)		(GPIO_GEN1) GPIO18	12
13	GPIO27 (GPIO_GEN2)		Ground	14
15	GPIO22 (GPIO_GEN3)		(GPIO_GEN4) GPIO23	16
17	3.3v DC Power		(GPIO_GEN5) GPIO24	18
19	GPIO10 (SPI_MOSI)		Ground	20
21	GPIO09 (SPI_MISO)		(GPIO_GEN6) GPIO25	22
23	GPIO11 (SPI_CLK)		(SPI_CE0_N) GPIO08	24
25	Ground		(SPI_CE1_N) GPIO07	26
27	ID_SD (I²C ID EEPROM)		(I²C ID EEPROM) ID_SC	28
29	GPIO05		Ground	30
31	GPIO06		GPIO12	32
33	GPIO13		Ground	34
35	GPIO19		GPIO16	36
37	GPIO26		GPIO20	38
39	Ground		GPIO21	40

1 – 3.3V a LDR1

2 – LDR2 a GPIO04 y condensador

3- condensador negativo a GND

RASPBERRY PI – DETECCION DE LUZ CON LDR – DESARROLLO

```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)

#define the pin that goes to the circuit
pin_to_circuit = 7

def rc_time (pin_to_circuit):
    count = 0

    #Output on the pin for
    GPIO.setup(pin_to_circuit, GPIO.OUT)
    GPIO.output(pin_to_circuit, GPIO.LOW)
    time.sleep(0.1)
```

```python
    #Change the pin back to input
    GPIO.setup(pin_to_circuit, GPIO.IN)

    #Count until the pin goes high
    while (GPIO.input(pin_to_circuit) == GPIO.LOW):
        count += 1

    return count

#Catch when script is interupted, cleanup correctly
try:
    # Main loop
    while True:
        print(rc_time(pin_to_circuit))
except KeyboardInterrupt:
    pass
finally:
    GPIO.cleanup()
```

RASPBERRY PI – MEASURING THE LIGHT – RESULT