

# Device Interfacing with Python and ZIO

vijaykumar@zillogic.com  
Zilogic Systems



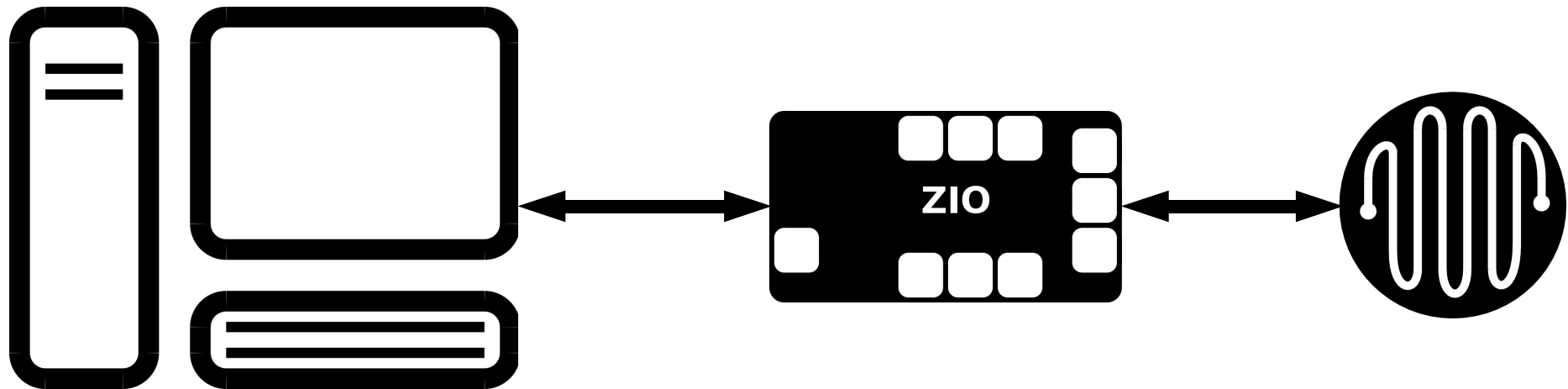
# Overview

- Introduction to ZIO
- Interfacing Devices
- Demo Projects

# Device Interfacing

- Parallel Port
- Limitations
  - Only Digital IO
  - Phased out

# ZIO Architecture



PC – The Universal Platform  
API to communicate with ZIO

USB based IO board

- Digital I/O
- Analog Input
- PWM Output
- I<sup>2</sup>C Bus
- SPI Bus

Sensors  
Transistors  
Relays  
LEDs  
Switches  
I<sup>2</sup>C Devices  
DC Motors  
IR receivers  
...

# ZIO Agent

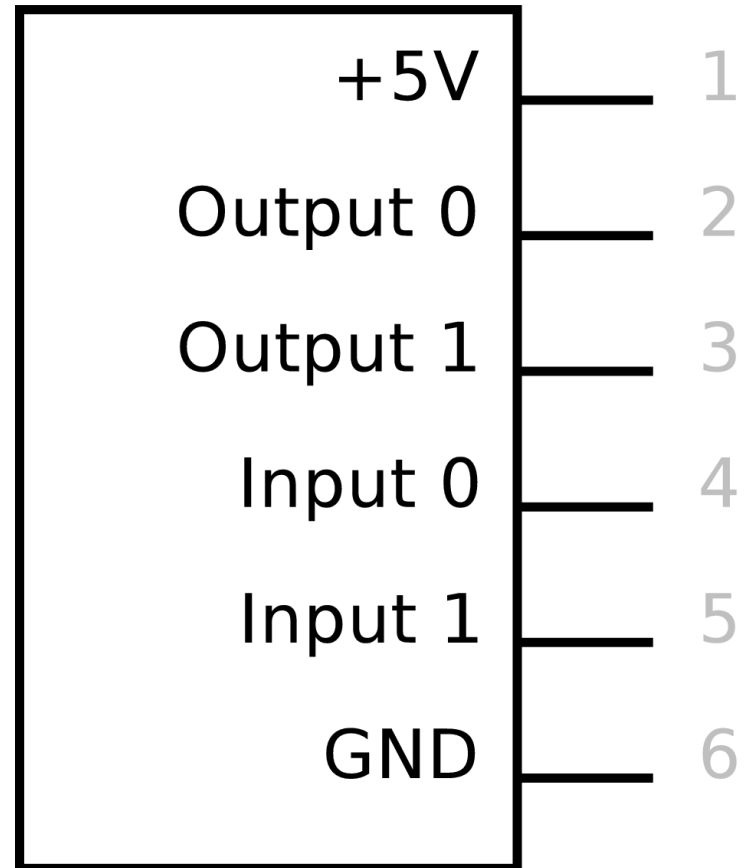
- ZIO Motherboard is powered by a ARM processor.
- ZIO Agent
  - Receives commands from PC through USB
  - Manipulates the interfaces based on the commands

# Ports

- GPIO
  - LEDs, Relays, Switches, MOSFETs, Optocouplers, ...
- Sensor
  - Temperature, Potentiometer, Light, Pressure, Humidity, ...
- PWM
  - DC Motor, Servo Motor, LED Brightness Control, ...
- I2C/SPI
  - RTCs, LCDs, IR Receivers, Sensors, Phone Line Interface
  - ...

# Ports (Contd.)

- Each Port has 6 signals
- Example GPIO port
  - +5V Power
  - GND
  - 2 Outputs
  - 2 Inputs
- Terminated in RJ12 connector

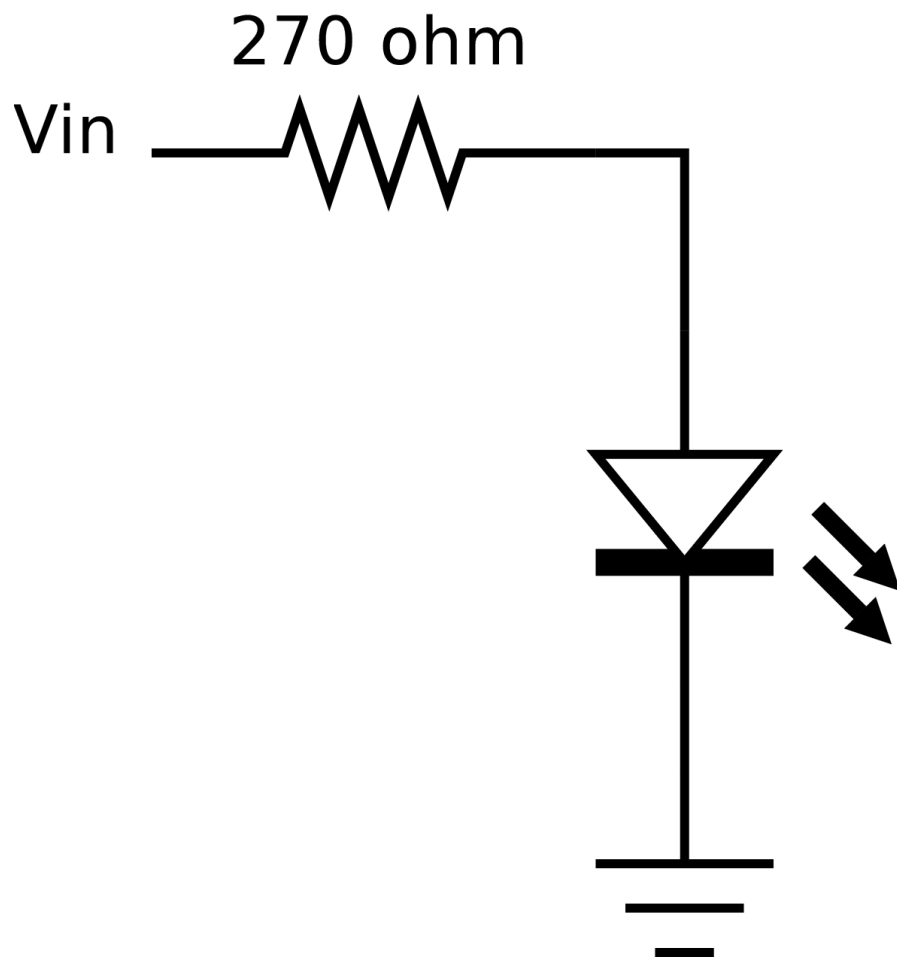


# Demo Board

- ZIO Motherboard
- Bread Board
- RJ12 Breakout Board
- Devices Board
- Temp. Sensor Board

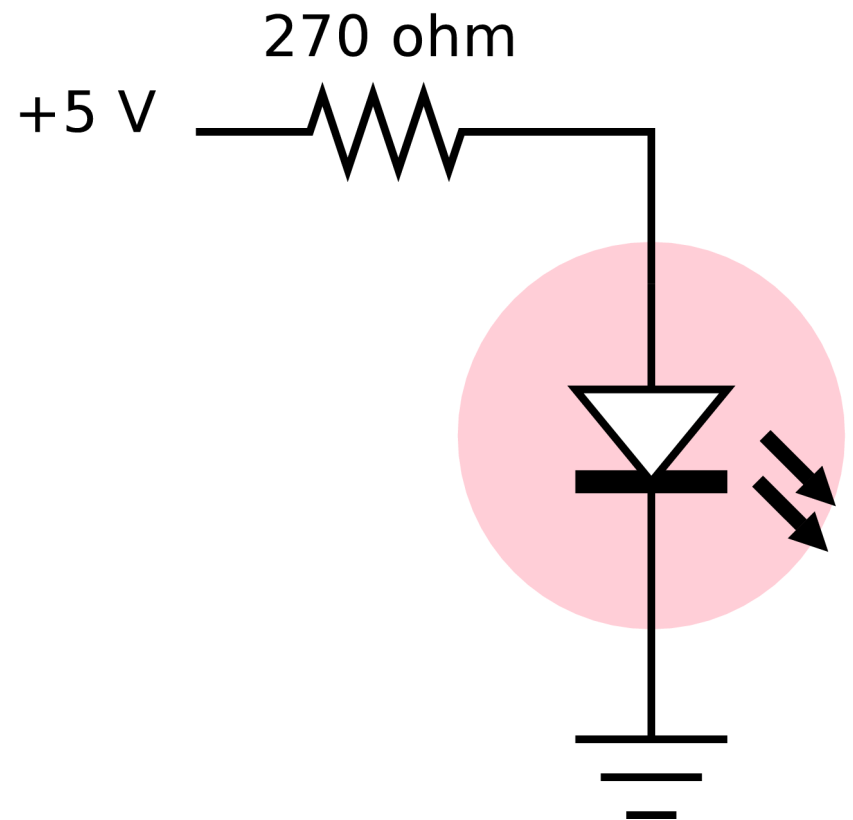
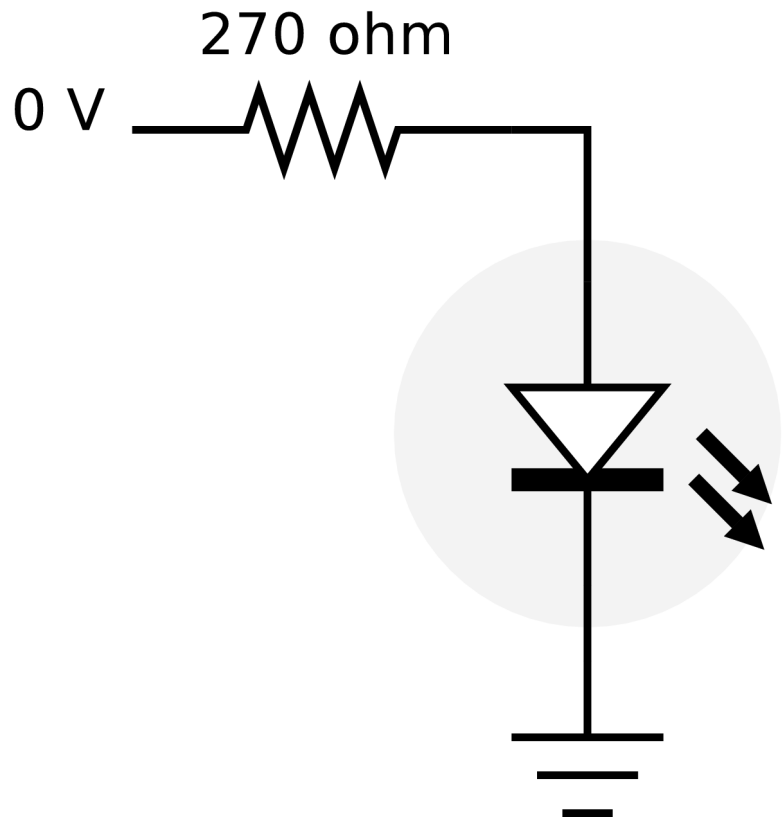


# LED

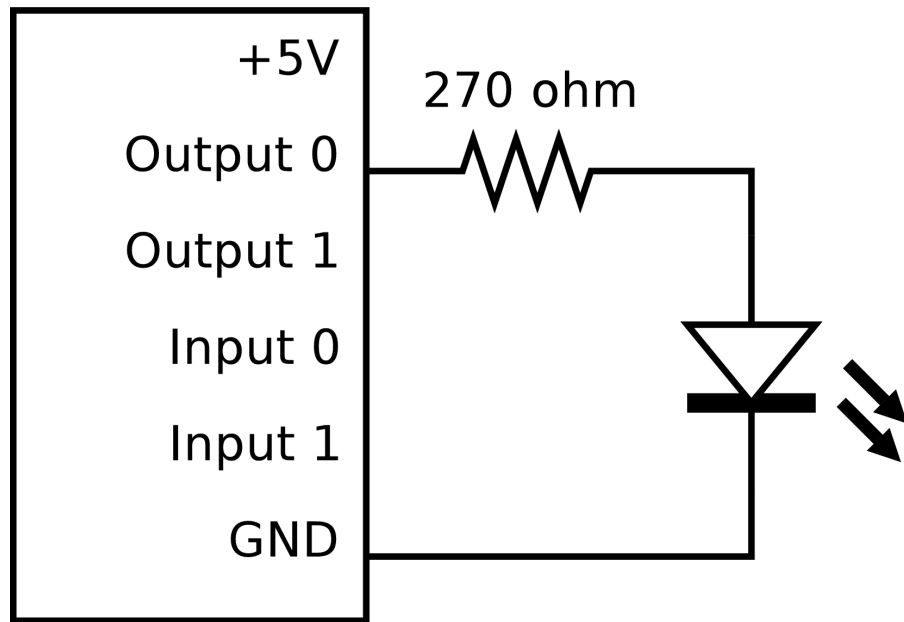


- Simple output devices
- Used for status indication, displays, lighting ...

# LED (Contd.)

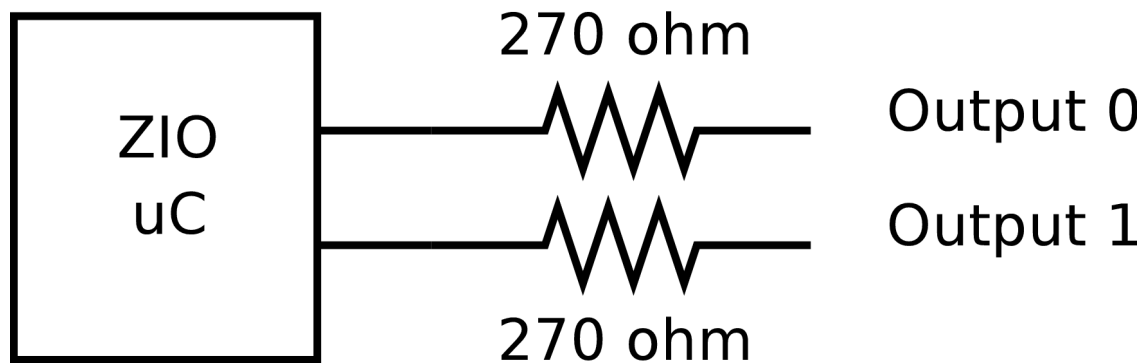


# Interface LED to ZIO



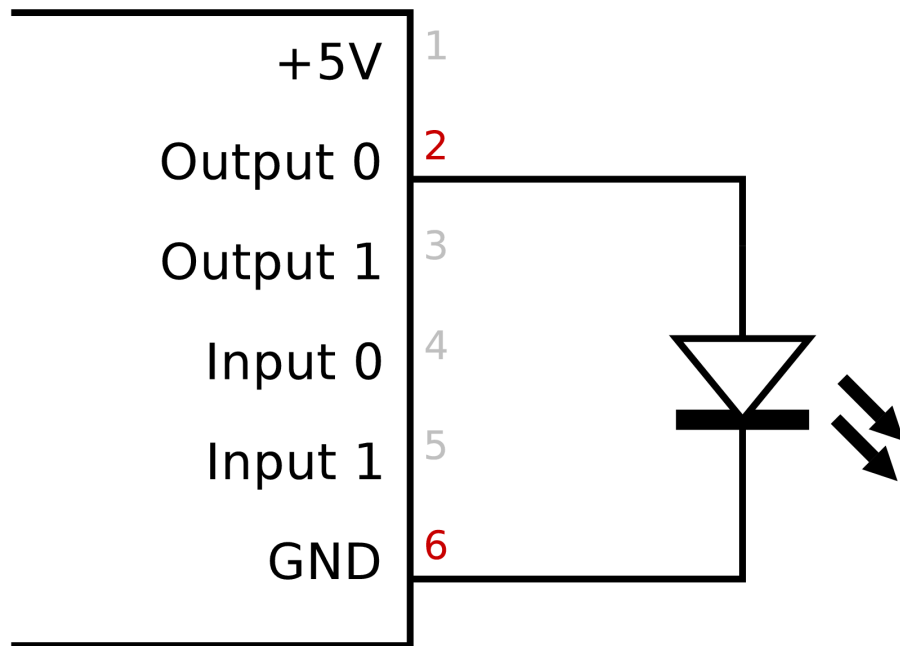
- GPIO Port
- Signals
  - 2 Outputs, 2 Inputs
  - +5V Supply, GND
- Setting Output to True, outputs 5V
- Setting Output to False, outputs 0V

# Interface LED to ZIO (Contd.)



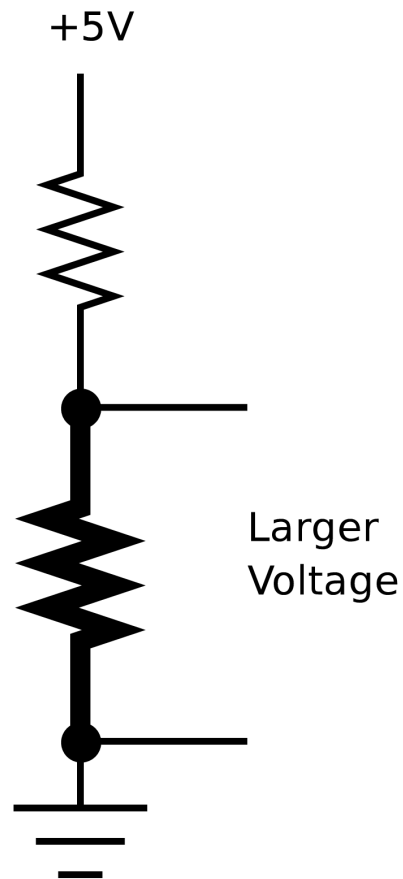
- GPIO outputs have a built-in series resistor
- Eliminates series resistors on external circuit

# Interface LED to ZIO (Contd.)

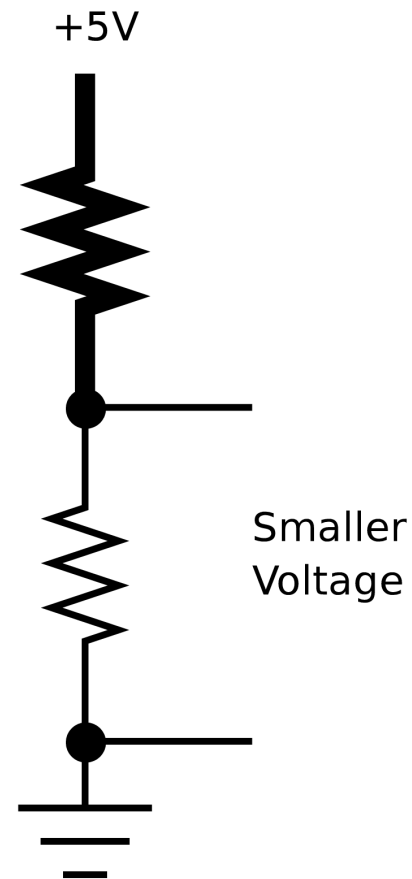


```
from zio import *  
agent = Agent("/dev/ttyUSB0")  
gpio = GPIO(agent)  
gpio.write_output_pin(0, True)  
gpio.write_output_pin(0, False)
```

# Voltage Divider



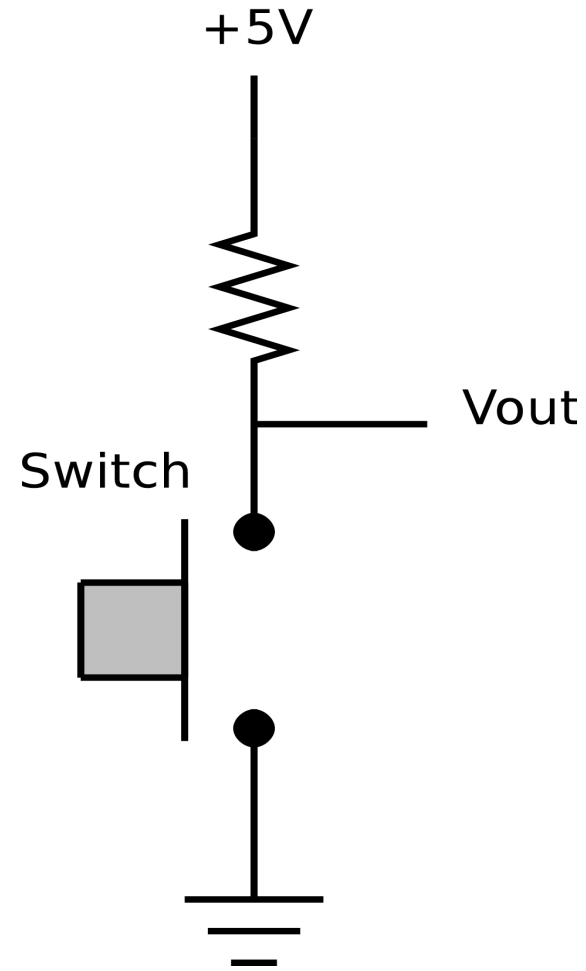
Larger the bottom Resistor  
Larger the Voltage



Smaller the bottom Resistor  
Smaller the Voltage

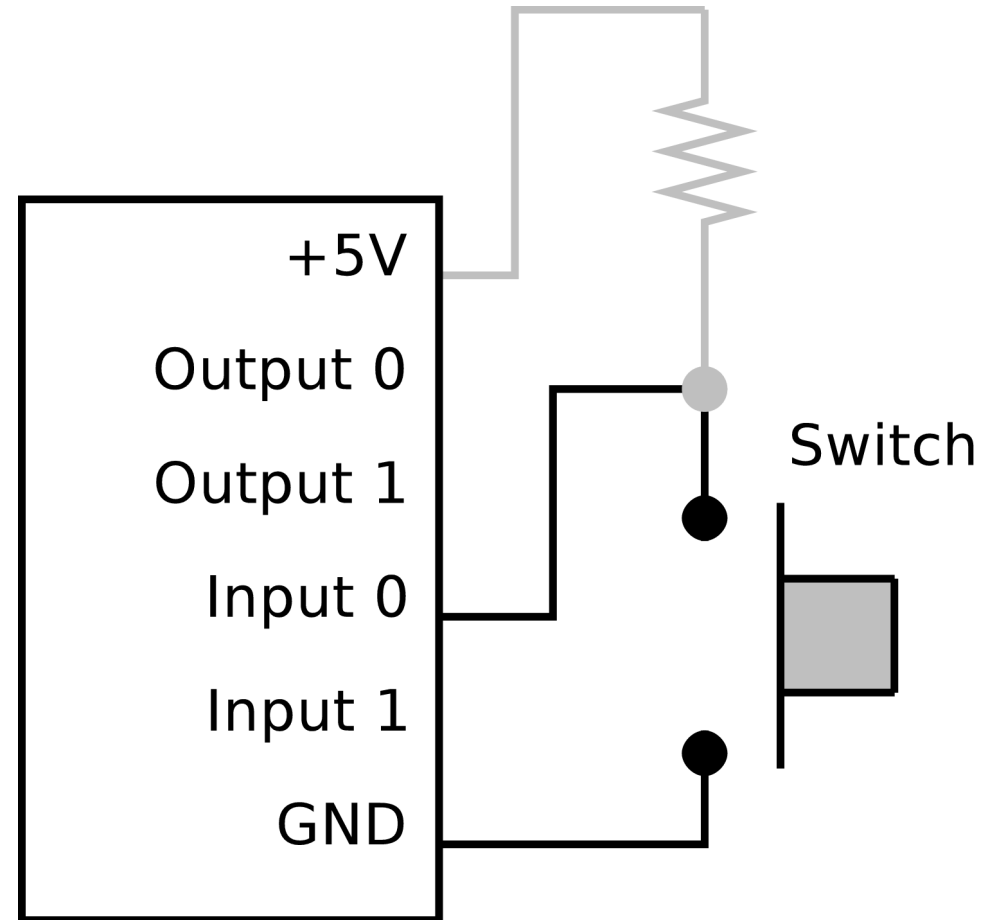
# Switch

- Simple input device
- Switch is closed
  - $V_{out} = 0V$
- Switch is open
  - $V_{out} = 5V$
- Switch state can be determined, by measuring  $V_{out}$ .



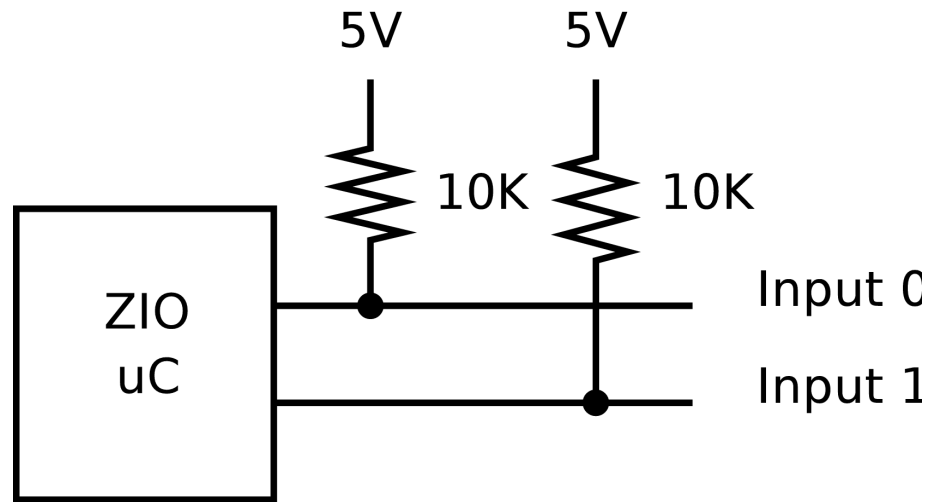
# Interface Switch to ZIO

- GPIO Input signals can test for a 0V or 5V.
- Input  $> 2V$ 
  - Read as True
- Input  $< 0.8V$ 
  - Read as False



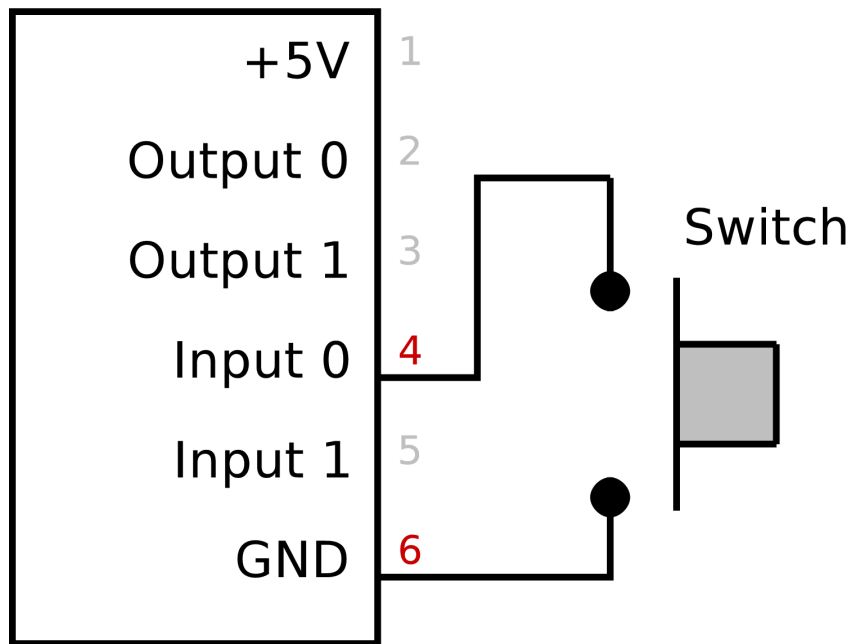


# Interface Switch to ZIO (Contd.)



- GPIO inputs have built-in pull-ups resistors
- Eliminates pull-ups on external circuits

# Interface Switch to ZIO (Contd.)

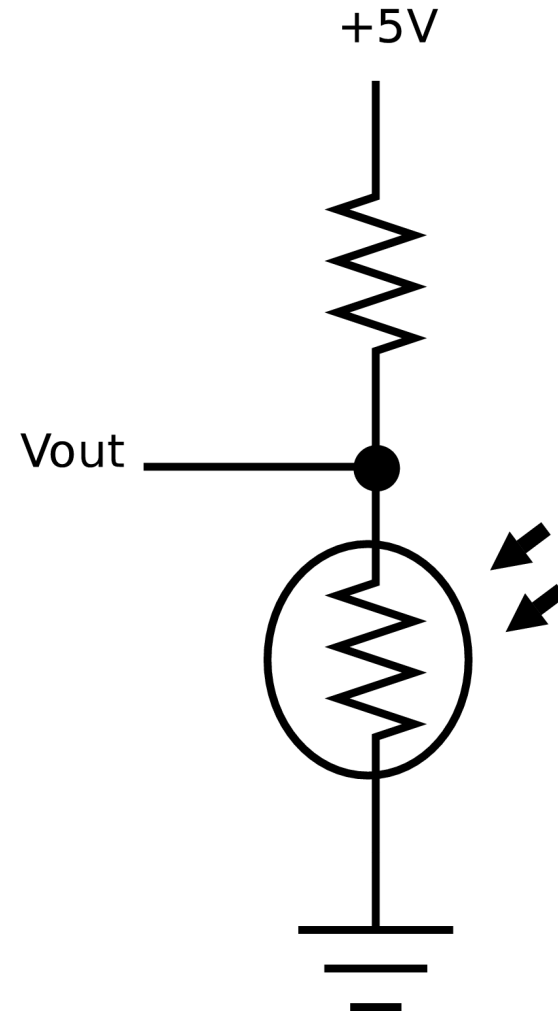


```
import time
from zio import *

agent = Agent("/dev/ttyUSB0")
gpio = GPIO(agent)
while True:
    print gpio.read_input_pin(0)
    time.sleep(1)
```

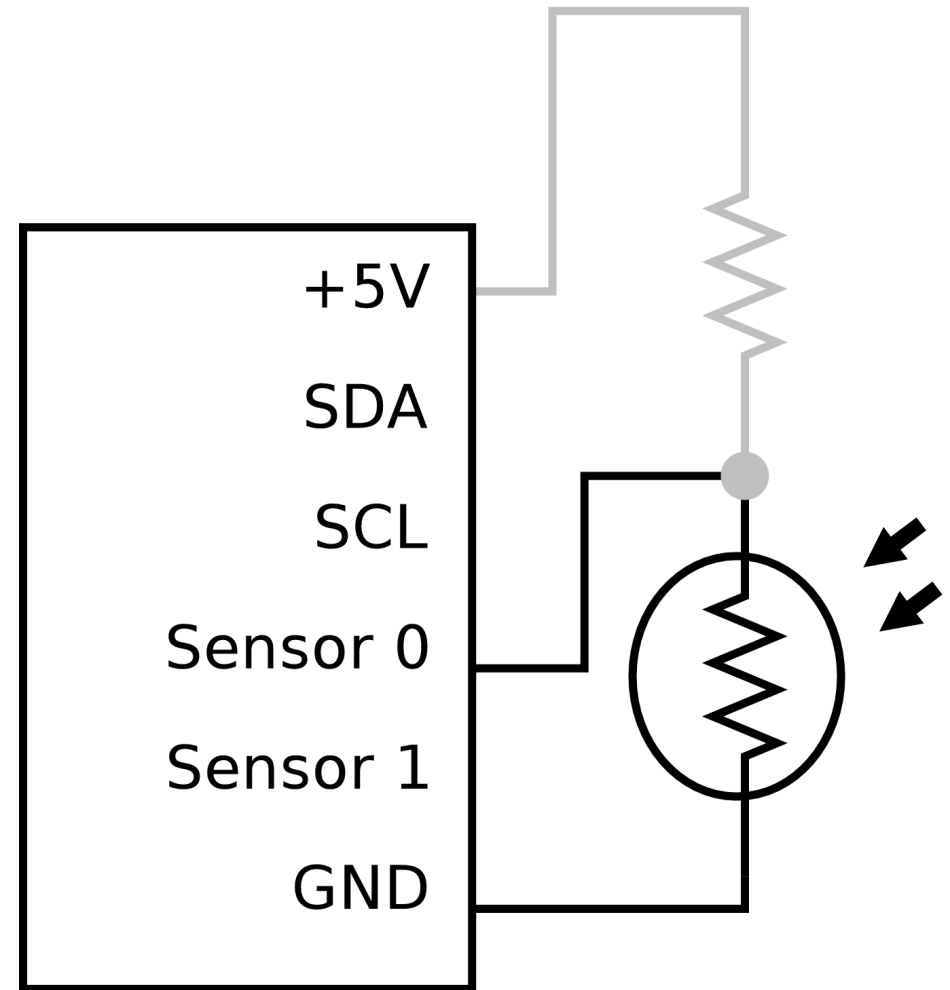
# Light Sensor (LDR)

- LDR – Light Dependent Resistor
- Resistance decreases with increase in light intensity
- Voltage  $V_{out}$  decreases with increase in light intensity

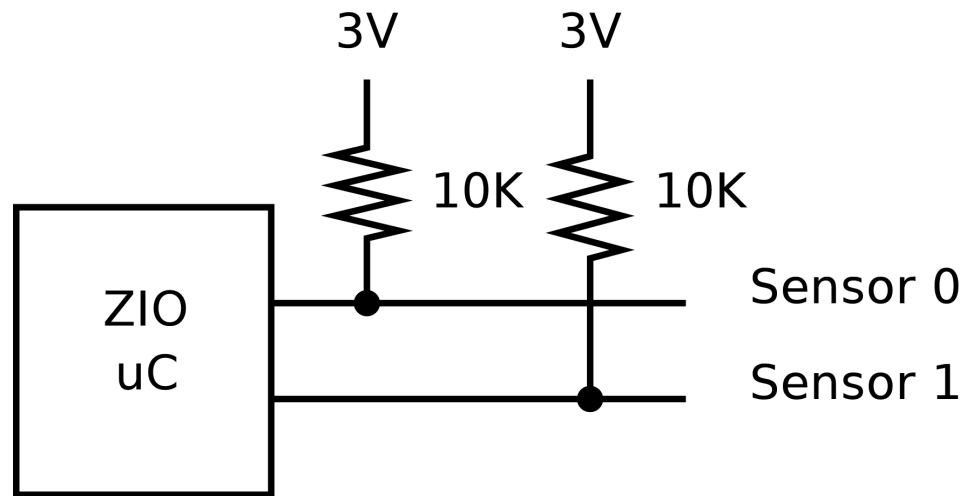


# Interface LDR to ZIO

- Sensor port
  - measure voltages between 0 – 3V
- Signals
  - +5V, GND
  - 2 Sensor Inputs
- Read the voltage at Sensor 0

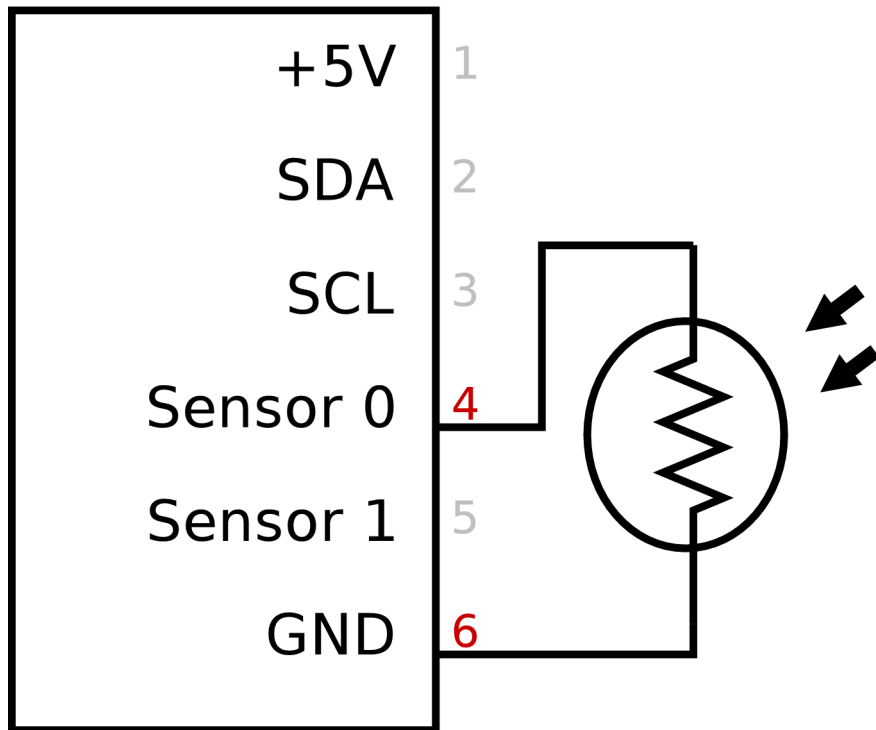


# Interface LDR to ZIO (Contd.)



- Sensor inputs have built-in pull-up resistors
- Eliminates pull-ups on external circuits
- Pull-ups connected to 3V, the max voltage that can be measured by sensor port.

# Interface LDR to ZIO (Contd.)

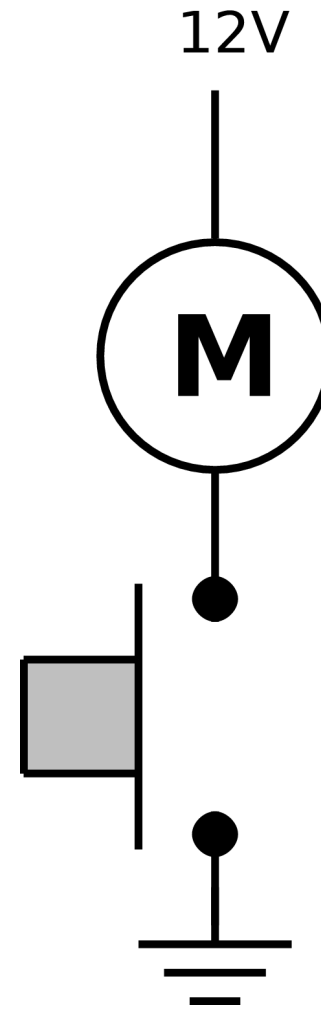


```
import time
from zio import *

agent = Agent("/dev/ttyUSB0")
sensor = Sensor(agent)
while True:
    print sensor.read_pin(0)
    time.sleep(1)
```

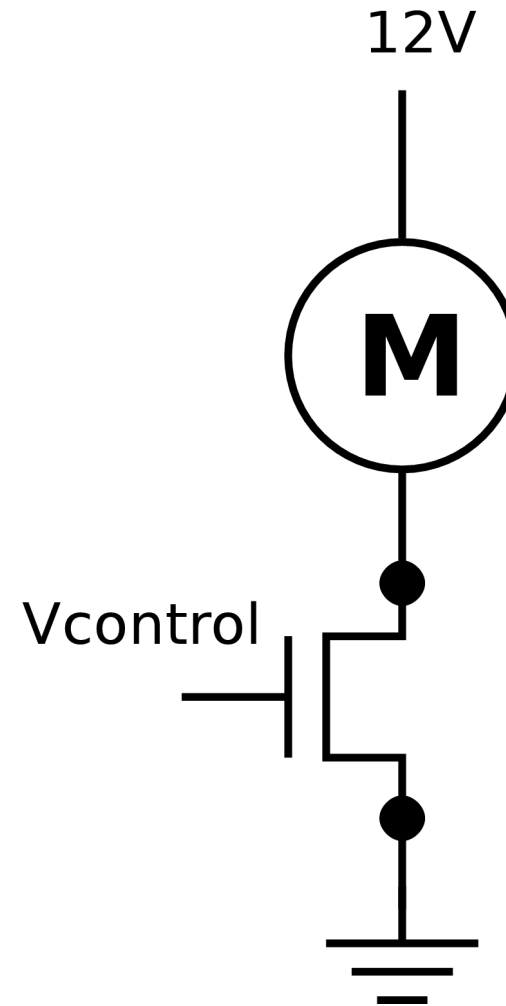
# DC Motor

- Examples
  - CPU Fan
  - Wheels of a Robot
  - CDROM drives
  - Printers
- DC motor controlled by a human operated switch



# DC Motor (Contd.)

- Replace switch by a MOSFET
- $V_{\text{control}} = 5\text{V}$ 
  - Motor turns ON
- $V_{\text{control}} = 0\text{V}$ 
  - Motor turns OFF

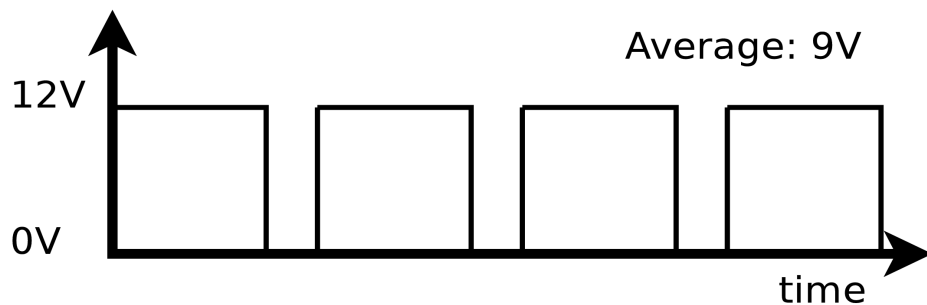
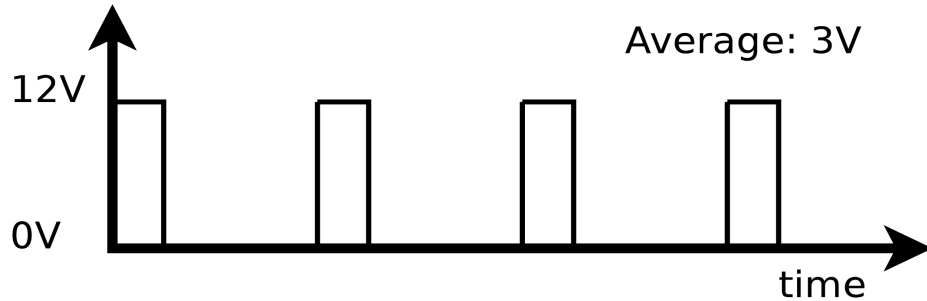
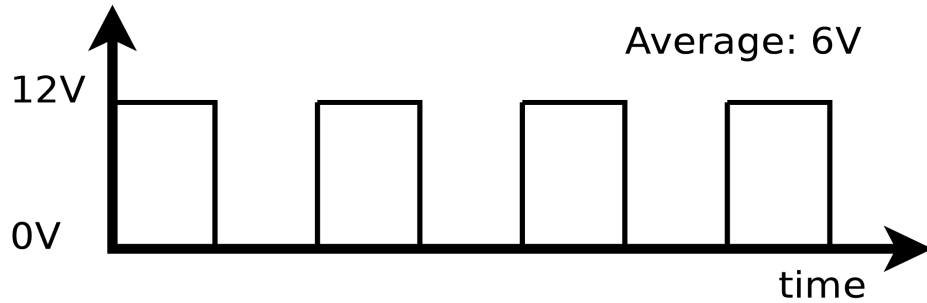




# DC Motor (Contd.)

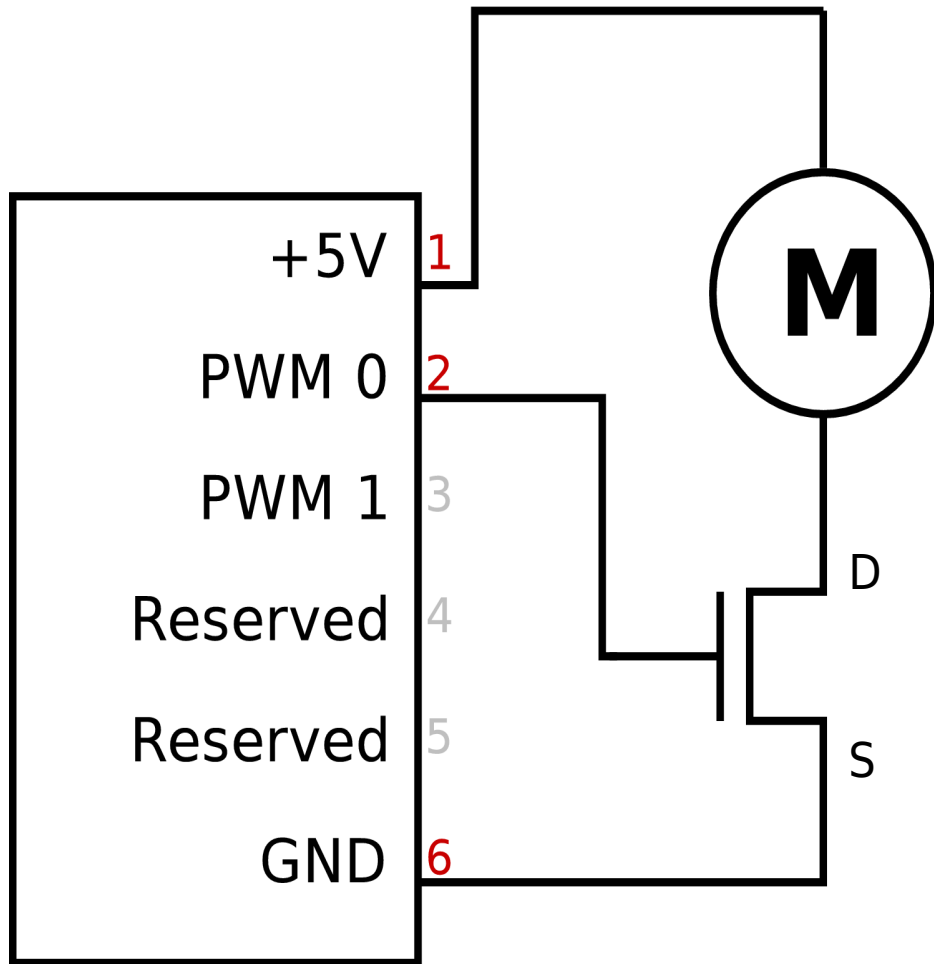
- GPIO port - motor ON and OFF
- Motor speed can be controlled
- DC motor speed is proportional to the supply voltage
- Speed control can be achieved by varying the averaging voltage delivered to the motor

# DC Motor (Contd.)



- Rapidly turn motor on and off
- Duty cycle
  - $(\text{ON time} / \text{Period}) * 100$
- Duty cycle 100%
  - Average voltage - 12V
- Duty cycle 50%
  - Average voltage - 6V

# Interface DC Motor to ZIO



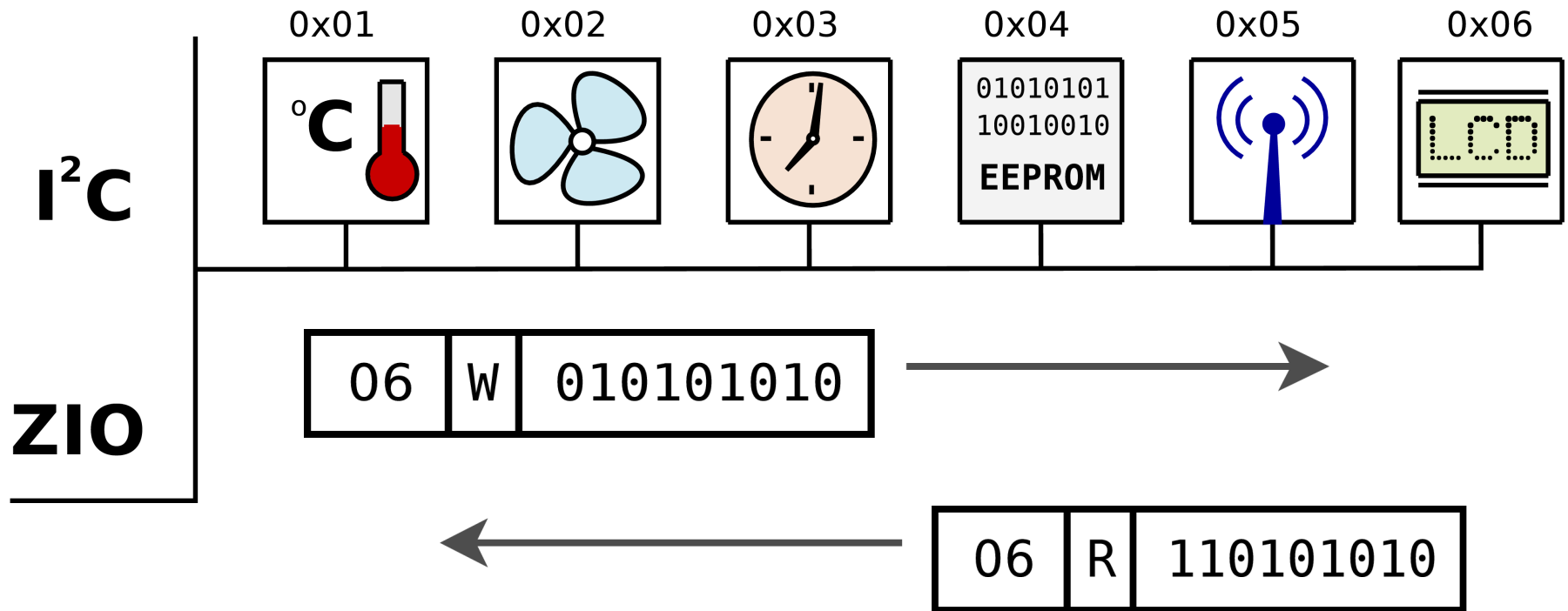
```
pwm = PWM(agent)
pwm.set_freq([0], 25)
pwm.set_duty([0], 100)
pwm.start([0])
```

```
pwm.set_duty([0], 50)
pwm.set_duty([0], 25)
```

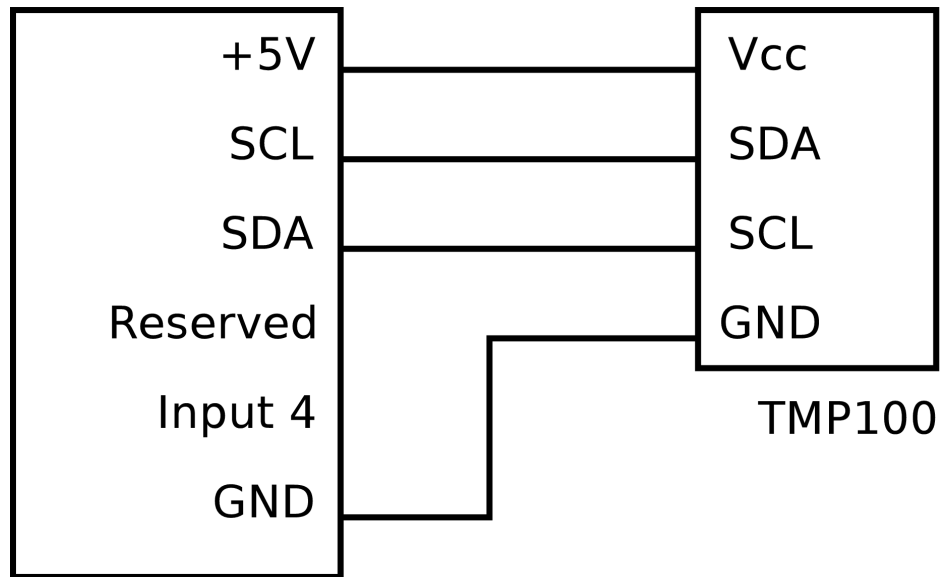
# Temperature Sensor

- Temperature Sensors
  - Resistive Sensors
  - Non-ratiometric Sensors
  - I2C / SPI Sensors
- I2C
  - kind of very simplified USB
  - connect devices to CPU
  - EEPROMs, RTCs, Accelerometers, Sensors ...

# I2C Bus



# Temperature Sensor



```
i2c = I2C(agent)
i2c.config(100)
while True:
    temp = i2c.read(0x48, 1)
    print temp[0]
    time.sleep(1)
```

# Demo Projects

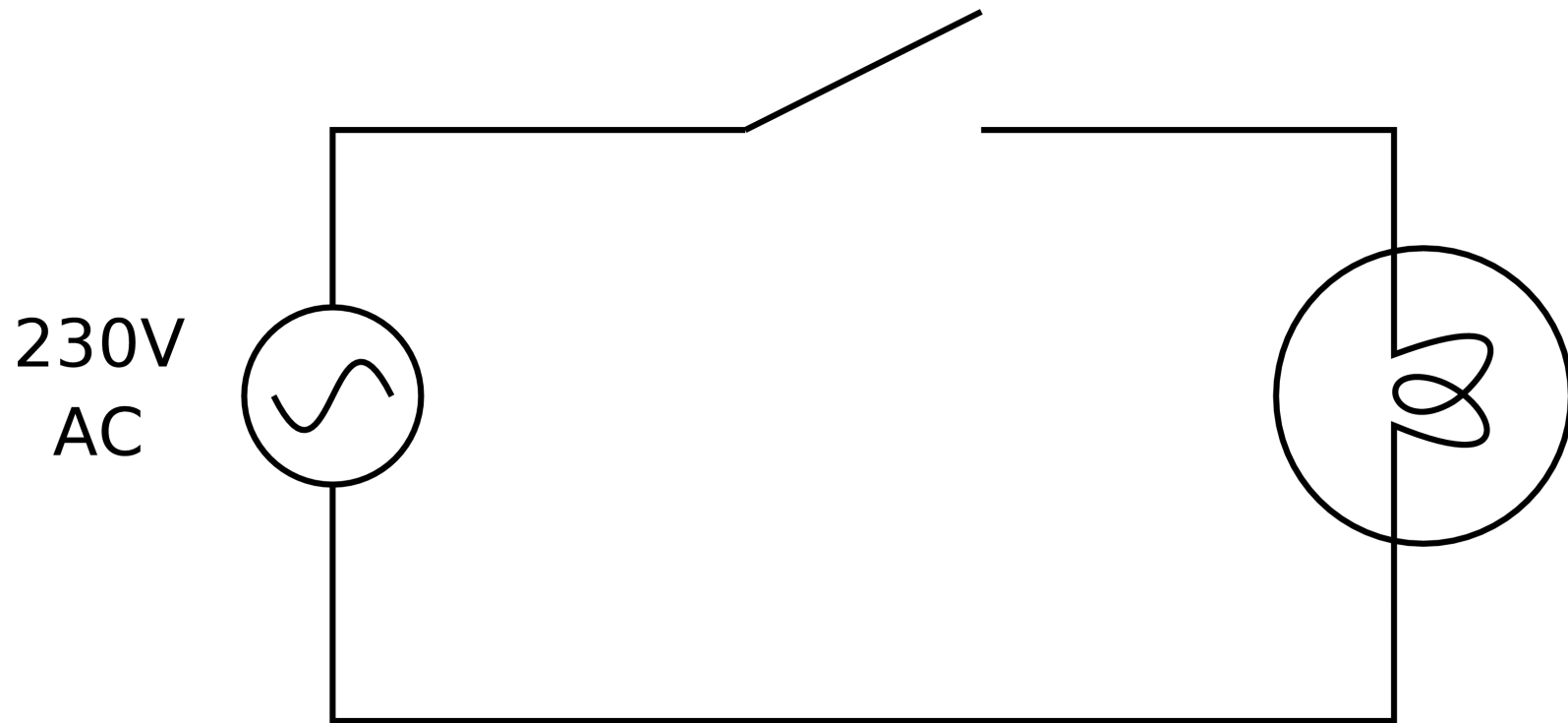
- Laser Pointer Presentation Control
- Light Bulb Control

# Laser Pointer Demo

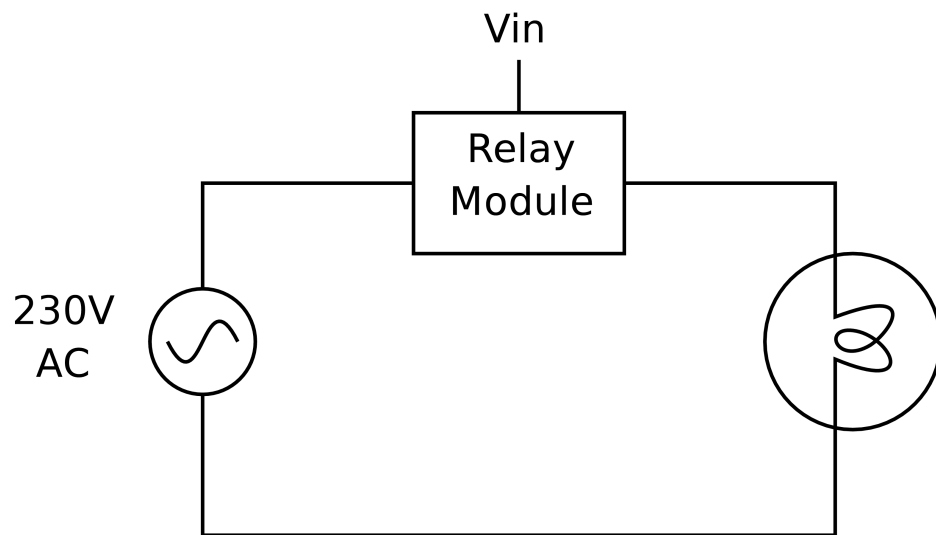
- Control presentation with input from the laser pointer.
- ZIO + LDR + Laser Pointer + Software Magic
- User shines laser on the LDR
- Software detects drop in the input voltage
- Software generates a key (Space) to active window (the presentation)



# Controlling a Light Bulb



# Controlling a Light Bulb (Contd.)



- Relay is a mechanical switch controlled by an electro magnet
- If  $V_{in} = 0V$  then bulb turns off
- If  $V_{in} = 5V$  then bulb turns on

# Questions

---

# Credits

- Behind the scenes Zilogic Team
  - PG <pg AT zilogic DOT com>
  - Kannan <kannan AT zilogic DOT com>
  - Development of Demo boards, Add-ons, ...
- Software Tools
  - Dia
  - Open Office