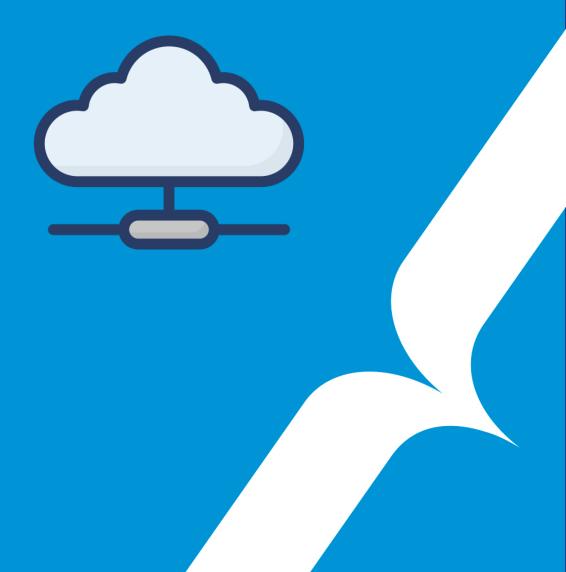# WEATHER

SESSION 1 - DRIVER DEVELOPMENT

WEATHER

TSL2561 Driver Development Subject (Python with smbus2)

Overview

In this project, you will learn to set up I²C communication between a Raspberry Pi and the TSL2561 light sensor. Through a series of exercises, you'll write Python functions using the `smbus2` module to control the sensor, read data, and calculate lux levels. Refer to the datasheet for details and use your problem-solving skills to write effective functions.

Part 1 : Setup

Exercise 1 : Introduction to Raspberry Pi, GPIO, and I²C

— **Objective** : Understand the basics of Raspberry Pi, GPIO (General Purpose Input/Output), and I²C (Inter-Integrated Circuit).
— **Tasks** :
1. Research GPIO and how it facilitates communication with external devices.
2. Learn the I²C protocol's purpose, especially in the context of sensors like the TSL2561.
3. Find the correct pinout on the Raspberry Pi for SDA and SCL.

Exercise 2 : Enabling I²C on the Raspberry Pi

— **Objective** : SSH into your Raspberry Pi and enable I²C communication.
— **Tasks** :
1. Connect to your Raspberry Pi via SSH.

{ EPITECH }

2. Use `sudo raspi-config` to enable I²C (navigate to *Interfacing Options > I2C*).
3. Confirm that I²C is enabled by listing available I²C modules.

Exercise 3 : Detecting the Sensor Address

— **Objective** : Locate the I²C address of your TSL2561 sensor.
— **Tasks** :
  1. Connect the TSL2561 sensor to the correct GPIO pins.
  2. Run `i2cdetect -y 1` to scan for connected devices.
  3. Note the sensor's address for use in your code.

Part 2 : Let's Code!

Here is the datasheet of the chipset

Exercise 4 : Understanding the Command Register

— **Objective** : Learn how the TSL2561's command register works.
— **Reference** : Datasheet, **pages 13-14**.
— **Tasks** :
  1. Study the command register layout in the datasheet, focusing on the command bit.
  2. Summarize how the command bit allows you to communicate with specific sensor registers.
  3. Document your findings as a reference.

Exercise 5 : Writing a Function to Power On the Sensor

— **Objective** : Write a function to power on the TSL2561.
— **Reference** : Datasheet, **page 14**.
— **Tasks** :
  1. Identify the control register address and the bit setting required to power on the sensor.
  2. Write a function using `smbus2` to send the power-on command.
  3. Confirm that the function works by reading the control register after execution.

{ EPITECH }

Exercise 6 : Writing a Function to Power Off the Sensor

— **Objective** : Write a function to power off the TSL2561.
— **Reference** : Datasheet, **page 14**.
— **Tasks** :
    1. Use the control register and the appropriate command to power off the sensor.
    2. Verify your function by reading the control register to ensure it is off.

Exercise 7 : Reading Sensor Data from Channels 0 and 1

— **Objective** : Access and read data from the TSL2561's two channels, CH0 and CH1.
— **Reference** : Datasheet, **pages 18-19**.
— **Tasks** :
    1. Find the data registers for CH0 and CH1 and identify their addresses and data lengths.
    2. Write functions to read 16-bit data from each channel.
    3. Log the data and analyze it to see if it aligns with expected values.

Exercise 8 : Calculating Lux from Channel Data

— **Objective** : Use the datasheet formulas to calculate lux (light intensity) and compare results.
— **Reference** : Datasheet, **page 23**.
— **Tasks** :
    1. Implement both lux calculation formulas provided in the datasheet.
    2. Compare the results of both formulas for accuracy and choose the more reliable one for your environment.

Exercise 9 : Setting the Sensor Sensitivity

— **Objective** : Write a function to set the sensor's gain (sensitivity).
— **Reference** : Datasheet, **page 20**.
— **Tasks** :
    1. Locate the timing register and determine the required values for different gain levels.
    2. Use this information to write a function adjusting the sensor's gain, then test it in various lighting conditions.

{ EPITECH }

BME280

Can you make a little driver for the BME280 now or are you going to use a all made lib > :)