IoT - modern applications

Measuring the intensity of light

Accurate and quantified light measurement is essential to create the desired results in practical everyday applications as well as in unique applications such as traffic lighting, poultry farming, horticulture, museum lighting, emergency exits, etc.

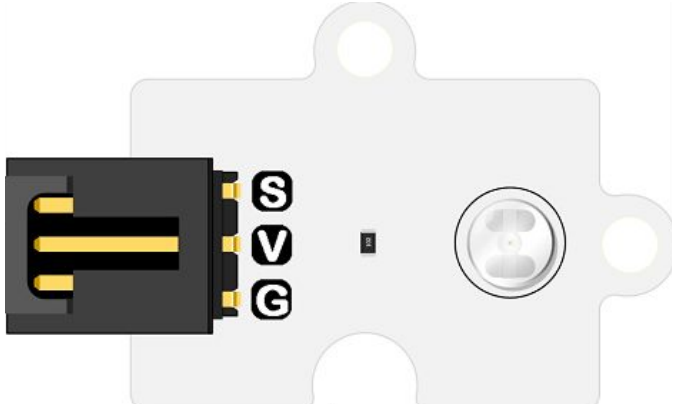
Therefore, measuring and analyzing light is an important step in ensuring efficiency and safety.

Measuring the intensity of light

Think of an example in your daily life that incorporates functions related to light intensity measurement

Measuring the intensity of light

The light sensor you have in your hand can give you the intensity of the light in the room.



Connect it to one of the sensor:bit analog connectors

Analogue terminals and ADC

The pico:ed with the RP2040 incorporates an analog-digital converter (ADC) that allows analog values to be read at specific pins.

The resolution of this ADC is 16bit and this practically means that by analyzing the analog input it can return a value from 0 to 65535, depending on the input voltage (0-3.3v).

The analogio library

The circuitpython analogio library contains classes for providing access to analog IO.

The AnalogIn class allows us to read the analog value of a pin. It takes the terminal as initialization parameter (board library).

class analogio.AnalogIn(pin)

With the value method it allows us to read the value

apin.value()

Let’s measure the intensity of light

Create a program that will display on the pico:ed display and on the serial line the analog value of the light intensity returned by the sensor, with a refresh rate of 5 sec.

Let’s measure the light intensity

STEM education LLC

A value (e.g. 41397) on the 0-65535 scale is not particularly understandable to the average user of a system. Can you convert the value to a 0-100 scale?

Recording of light intensity

As mentioned earlier, optimizing the efficiency of the applications around us is possible by measuring and analyzing these measurements.

The analysis can be achieved by collecting a series of data from which, after some processing, useful information will be extracted.

Data collection

In the previous lesson you kept a set of data as it was necessary to display the proximity on the pico:ed screen.

Do you have access to this data?

How could we keep our data permanently?

Working with files

File management is an important function in all programming languages and the relevant methods/functions are built into them.

In python the key function for file management is open()

It takes 2 parameters

- ➔ the file name
- ➔ the way to open the file

E.g. f = open('myfile.txt', 'w')

Working with files

- The basic ways to open a file are:
- "r" - opens a file for reading - if the file does not exist it returns an error
- "a" - opens a file for adding - if the file does not exist it creates it
- "w" - opens a file for writing - if the file does not exist creates it - if it exists deletes its contents and writes to it from the beginning
- "x" - creates a file - if the file exists returns error

Working with files

To write to a file we use the write() method

Prerequisite for write() to work is that the file must be opened with type "a" or "w"

As a parameter it takes an alphanumeric (string)

E.g. f.write("Hello world")

After we are done with the additions to our file and in order to save our changes, we call the close() method

E.g. f.close()

Measurement and recording of light intensity

Can you modify your program to save the 2-minute measurements in a file named Lmeasure.txt ?

Where do you think this file was stored?

Can you see its contents?

Recovery of light intensity data

To read the contents of a file, after opening the file with the "r" read mode, we can use the following methods:

read() returns the contents of the file - optionally takes an integer parameter indicating the number of characters to return

readline() returns the character set of a line in the file

Each subsequent read with one or the other method returns the next characters or lines in the sequence. After we have finished we must close the file with the **close()** method

Recovery and calculation of the average

Modify your program so that after saving the data to the file, it retrieves them in a list and calculates the average light intensity for that period of time.

Data processing

Collect the 3' time interval measurements

Analyse your data and find:

- → Find the average
- → The maximum value
- → The minimum value