

IoT Worksheet 3

Shadhan Sidique

Contents

Introduction.....	2
Requirements	2
Software Dependencies.....	2
Choice of Software.....	2
Node.js.....	2
Python	2
SQLite.....	2
Installation	2
Microbit	2
COM Port Setup.....	2
Runtime	3

Introduction

This documentation goes through a program that demonstrates a way to represent the Internet of Things. I have chosen to make a webserver that collects and stores temperature and light data from a BBC Micro:bit and generates graphs for it.

Requirements

Although this project is a small demonstration of what is possible through IoT, it can be used to do many more things using the serial data interface. In this documentation, I will go through the necessary steps for you to set this program up on your own. The hardware components required are:

- A computer (this hosts the webserver as well as reads the data through the serial interface)
- BBC Micro:bit device (this collects the data and sends it back through the serial interface)

Software Dependencies

The computer needs to have python (at least 3.9) as well as node.js (at least v18.12.1).

It is also required for you to install the required dependencies for node using the command “npm install” as well as for python using the command “pip install pyserial requests”. Note that the command must be run inside of the directory with the project for the node dependencies.

Choice of Software

Node.js

Node.js is used as the webserver as it is highly performant for network operations such as accepting and sending network requests as it has a non-blocking event loop which allows for asynchronous support even on a single thread.

Python

Python is used as the interface to read the serial data as it is an intuitive language to work and design with even for lower level applications such as reading bytes from a serial connection.

SQLite

For the database, SQLite is being used as it is a decent alternative to a full-fledged SQL database which includes support for multiple clients and data replicating.

Installation

Microbit

Connect the BBC Micro:bit to a computer and copy the hex file to it. If you want to make changes to the code, you can use <https://python.microbit.org/> in order to make the hex file using the source python code.

COM Port Setup

Once the BBC Micro:bit has been programmed properly and is connected to the computer, you can determine the correct COM port to use by running “python serial/list_com_ports.py”.

```
C:\Users\Shadhan Sidique\Documents\microbit> python serial/list_com_ports.py
COM5 - Standard Serial over Bluetooth link (COM5)
COM3 - Standard Serial over Bluetooth link (COM3)
COM7 - microbit

C:\Users\Shadhan Sidique\Documents\microbit>
```

Figure 1: List of COM ports

Once the COM port for the BBC Micro:bit is determined, open the file “serial/read.py” and edit line number 10 with the correct COM port.

Runtime

Assuming the BBC Microbit is plugged in and the COM port is assigned correctly, running the file named “start.cmd” should start up both the webserver as well as the program that reports the data values assuming that they are coming through.

To get a graph of the data, you can access the webserver at <http://localhost:3000> on the machine running the webserver, and the graph is shown for all the accumulated data at the endpoint <http://localhost:3000/report/temp> where temp stands for temperature, and at <http://localhost:3000/report/light> for light.

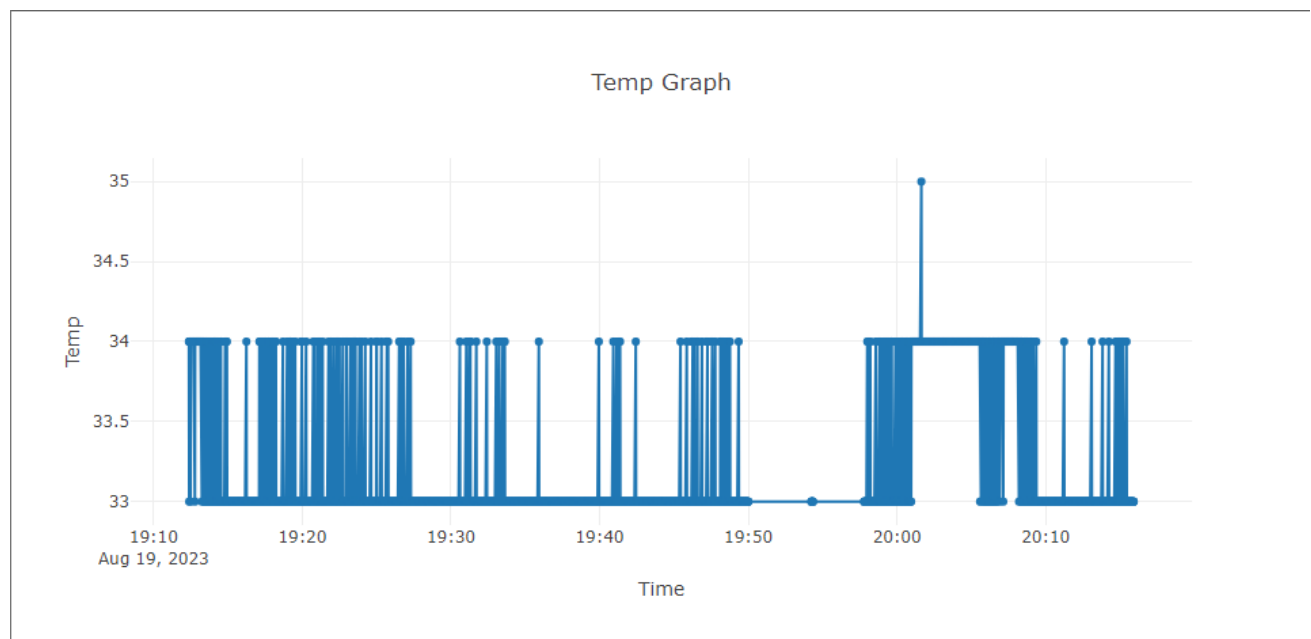


Figure 2: Temperature Graph

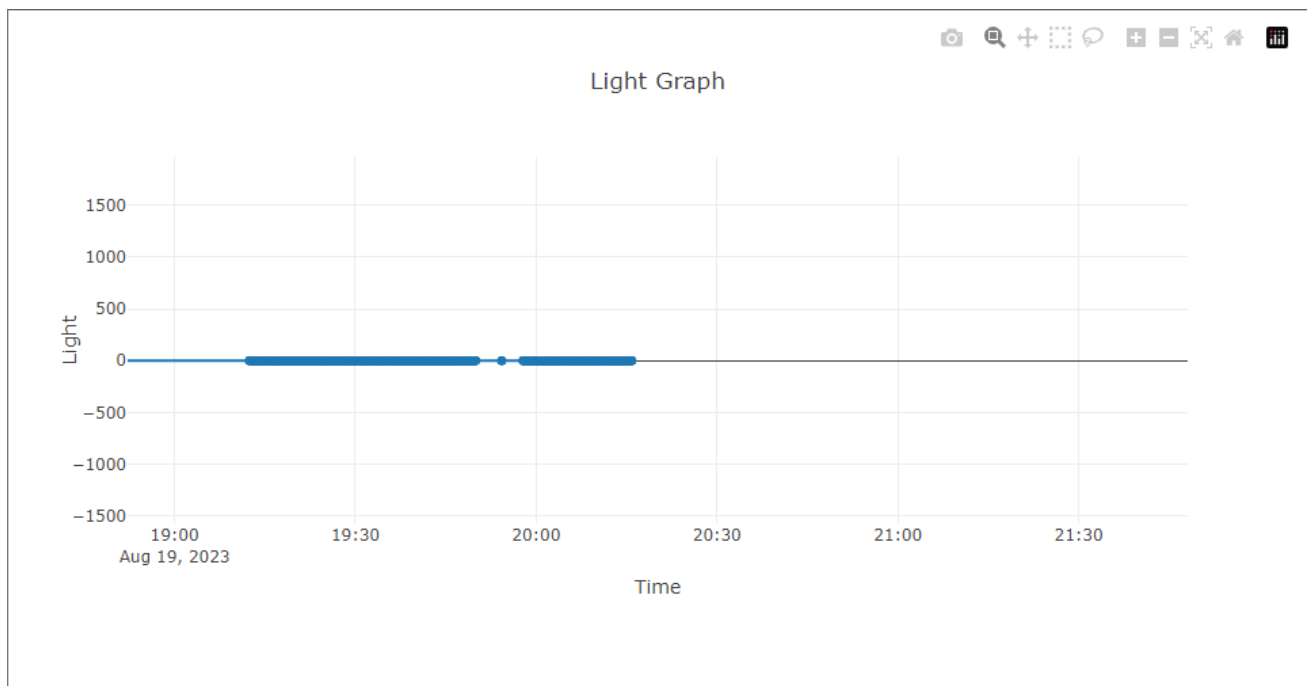


Figure 3: Light Graph

Note: The light sensor keeps sending back a light level of 0, even throughout a long duration.