|  |  |
| --- | --- |
| **Thành viên** | **Nội dung thực hiện** |
| Trần Thanh Hùng | Câu 1 (công việc - 100%) |
| Nguyễn Trung Hiếu | Câu 2 (công việc - 100%) |
| Trần Nguyễn Việt Hoàng | Câu 3 (công việc - 100%) |

**BẢNG PHÂN CHIA CÔNG VIỆC:**

**ĐÁNH GIÁ KHÁC (\*):**

|  |  |
| --- | --- |
| **Nội dung** | **Kết quả** |
| Tổng thời gian thực hiện bài thực hành trung bình (1) | 1 tuần |
| Link Video thực hiện (2)  *(nếu có)* | <https://drive.google.com/file/d/1Y4a7PSebQAqCVZiWRwM9OS5Vowv7U1vu> |
| Ý kiến (3) *(nếu có)*  + Khó khăn  + Đề xuất … |  |
| Điểm tự đánh giá (4) | 9/10 |
| (\*): phần (1) và (4) bắt buộc thực hiện. | |

GVHD: **Phan Trung Phát**

Nhóm: **TH2.02**

|  |  |
| --- | --- |
| **Họ và tên** | **MSSV** |
| Trần Nguyễn Việt Hoàng | 23520541 |
| Trần Thanh Hùng | 23520580 |
| Nguyễn Trung Hiếu | 23520487 |

**Lập trình TelosB với TinyOS**

5

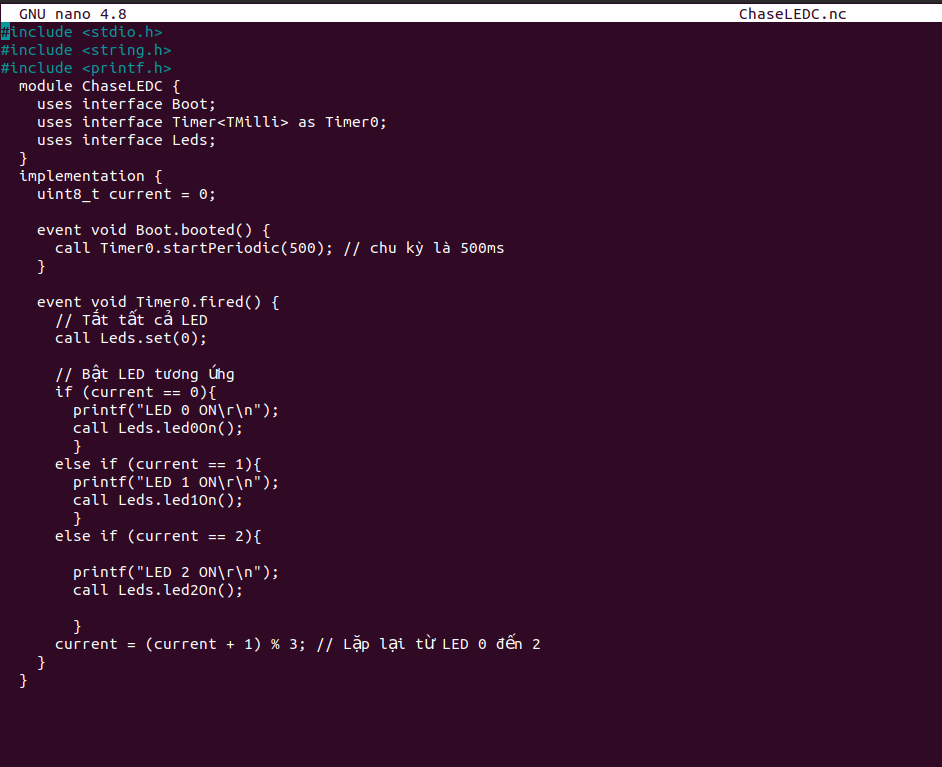
# Câu hỏi 1. Tìm hiểu thư viện printf trong TinyOS để gửi thông tin từ TelosB sang máy tính. Cập nhật lại chương trình để TelosB gửi thông tin cho biết đèn số mấy đang sáng.

## Minh Chứng

A computer screen with white text

AI-generated content may be incorrect.

1. Mã nguồn ChaseLEDAppC.nc

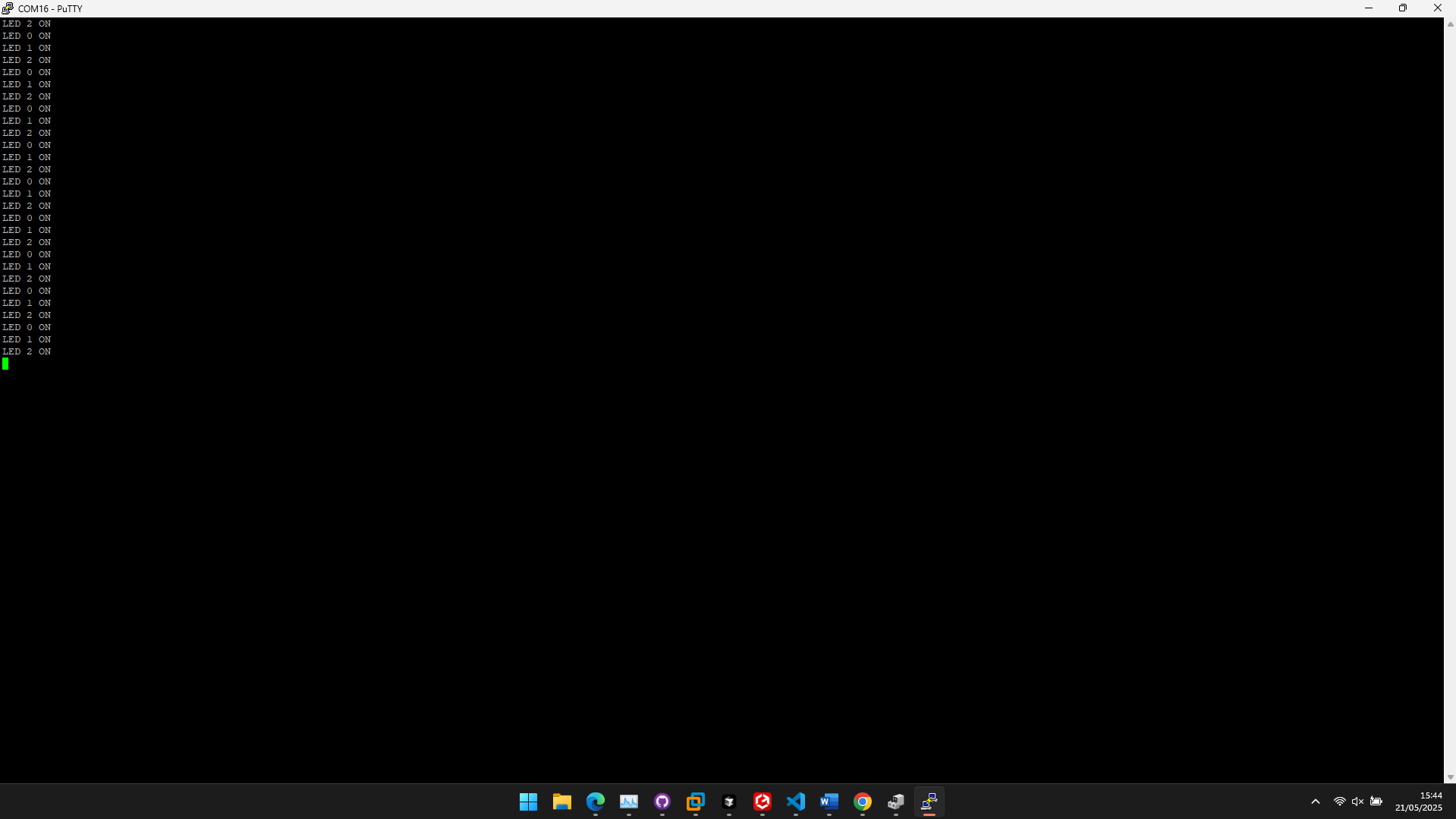


1. Mã nguồn của ChaseLEDC.nc

A computer screen with white text

AI-generated content may be incorrect.

1. Makefile sau khi sửa



1. Kết quả

## Kết quả:

Sau mỗi 500ms, các đèn trên TelosB sẽ thay đổi lần lượt từ đỏ, vàng, xanh dương và đồng thời gửi thông tin đèn đang sáng về máy tính thông qua Serial để thông báo.

Đỏ 🡪 LED 0 ON

Vàng 🡪 LED 1 ON

Xanh à LED 2 ON

## Giải thích

Tại file ChaseLedAppC.nc, thêm một components mới SerialPrintfC để có thể gửi dữ liệu thông qua Serial Port.

Tại file ChaseLedC.nc thêm các phần:

Include các thư viện cần thiết:

stdio.h: xử lý nhập xuất

string.h: xử lý chuỗi

printf.h: xử lý nhập xuất thông qua cổng serial

Tại các lệnh điều kiện xử lý điều kiện để bật LED thêm các dòng printf(); để gửi thông tin về đèn nào đang sáng. Đồng thời kết thúc chuỗi thêm \r\n nhằm đưa con trỏ về đầu dòng và xuống hàng, vì sử dụng putty trên hệ điều hành Windows để đọc dữ liệu từ Serial nên sẽ phải kết hợp \r\n để dữ liệu xuất ra chuẩn hơn.

Tại file Makefile thêm:

PFLAGS += -I$(TINYOS\_ROOT\_DIR)/tos/lib/printf: thêm đường dẫn của thư viện printf trong TinyOS vào danh sách mà trình biên dịch tìm kiếm file header khi biên dịch.

# Câu hỏi 2. Viết chương trình cho TelosB để đọc dữ liệu cảm biến ánh sáng và biểu diễn thông qua các đèn LED.

**File LightSensorAppC (tệp cấu hình)**

A screenshot of a computer program

AI-generated content may be incorrect.

1. Mã nguồn LightSensorAppC

- Định nghĩa cách các thành phần được kết nối với nhau bao gồm

+ LightSensor as app : logic chính của đèn trên cảm biến ( đặt tên là App)

+ MainC: Khởi động ứng dụng.

+ LedsC: Điều khiển các đèn LED trên bo mạch.

+ TimerMilliC(): Bộ đếm thời gian (timer) với độ chính xác mili giây.

+ App.Boot -> MainC: Kết nối interface Boot mà App cần sử dụng với interface Boot được cung cấp bởi MainC. Khi hệ thống khởi động, MainC sẽ gọi sự kiện booted() trên App.Boot.

+ App.Leds -> LedsC: Kết nối interface Leds mà App cần sử dụng với interface Leds được cung cấp bởi LedsC. Để cho phép App gọi các lệnh điều khiển bật tắt đèn.

+ App.Timer -> TimerMilliC: Kết nối interface Timer mà App cần sử dụng với interface Timer được cung cấp bởi TimerMilliC. Để cho phép App khởi động timer và nhận sự kiện khi timer kích hoạt.

+ SerialPrintfC: In dữ liệu ngưỡng ánh sáng ra cổng serial.

+ HamamatsuS10871Tsrc() as LightSensor: Trình điều khiển cho cảm biến ánh sáng (đặt tên là LightSensor).

+ App.LightRead -> LightSensor: Kết nối interface LightRead mà App cần sử dụng với interface Read được cung cấp bởi LightSensor (HamamatsuS10871Tsrc). Để cho phép App yêu cầu đọc giá trị từ cảm biến ánh sáng.

**File LightSensor (tệp logic chính)**

A screenshot of a computer program

AI-generated content may be incorrect.

1. Mã nguồn LightSensor

* Khởi động: Khi mote TelosB được cấp nguồn hoặc reset, MainC khởi chạy kích hoạt sự kiện Boot.booted() trong module LightSensor (thông qua kết nối App.Boot -> MainC).
* Boot.booted(): Thực hiện đọc giá trị cảm biến ánh sáng mỗi giây.
* Timer.fired(): Yêu cầu driver của cảm biến ánh sáng (HamamatsuS10871Tsrc) đọc giá trị.
* Khi việc đọc hoàn tất, driver cảm biến kích hoạt sự kiện LightRead.readDone() trong module LightSensor, truyền vào kết quả (thành công/lỗi) và giá trị thô val.  
  + Nếu đọc thành công, giá trị thô val được chuyển đổi thành luminance (đơn vị lux) và được in ra cổng serial.  
  + Nếu luminance >1000 thực hiện tắt đèn xanh dương và vàng, bật đèn đỏ.

+ Nếu luminance >=100 và <= 1000 thực hiện tắt đèn xanh dương và đỏ, bật đèn vàng.

+ Nếu luminance < 100 thực hiện tắt đèn đỏ và vàng, bật đèn xanh dương.

**Makefile (tệp biên dịch)**

A computer code with white text

AI-generated content may be incorrect.

1. Makefile

Cung cấp các chỉ thị cho trình biên dịch TinyOS, xác định LightSensorAppC là component chính của ứng dụng, thiết lập các tham số bao gồm kích thước bộ đệm cho hàm printf, tốc độ truyền dữ liệu qua cổng serial (baud rate), và chỉ định đường dẫn đến thư viện printf.

A screenshot of a computer

AI-generated content may be incorrect.

1. Giá trị cảm biến ánh sáng được in ra

A computer with a computer chip attached to it

AI-generated content may be incorrect.

1. Đèn vàng khi ánh sáng bình thường (100lx <= Ánh sáng <=1000lx)



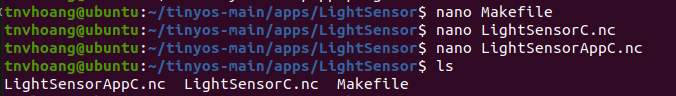
1. Đèn đỏ khi chiếu đèn vào cảm biến ánh sáng (Ánh sáng > 1000lx)



1. Đèn xanh dương khi che cảm biến ánh sáng (Ánh sáng < 100lx)

# Câu hỏi 3. Viết chương trình cho TelosB để đọc dữ liệu cảm biến ánh sáng và nhiệt độ, gửi dữ liệu sang máy tính.

Tạo thư cho chương trình đọc dữ liệu cảm biến ánh sáng và nhiệt độ LightSensor (chứa 3 file thực thi: Makefile, LightSensorAppC.nc và LightSensorC.nc)

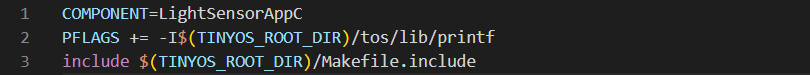


1. Tạo thư mục và file thực thi chương trình

## Viết chương trình

Makefile

* PFLAGS += -I$(TINYOS\_ROOT\_DIR)/tos/lib/printf: thêm đường dẫn đến thư viện printf của TinyOS vào cờ biên dịch để sử dụng hàm printf trong LightSensorC.nc



1. Thiết lập Makefile

Flie cấu hình LightSensorAppC.nc

* Khai báo
  + TimerMilliC as Timer0: tạo bộ định thời
  + PrintfC, SerialStartC: in dữ liệu qua cổng seria.
  + HamamatsuS1087ParC as LightSensor: cảm biến ánh sáng
  + SensirionSht11C as TempSensor: cảm biến nhiệt độ
* Kết nối giao diện
  + Time: giao diện Timer<TMilli>, sử dụng bộ định thời để thực hiện định kì mỗi 1 giây
  + LightRead và TempRead: giao diện Read<uint16\_t>, dùng để đọc cảm biến ánh sáng và nhiệt độ

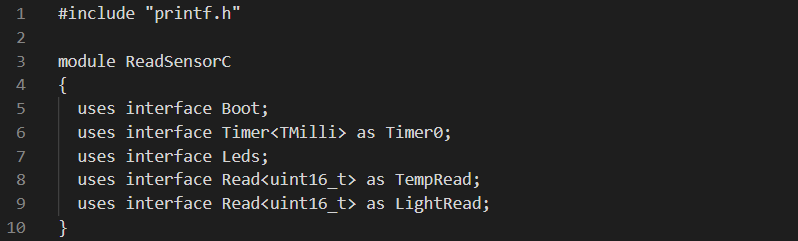
A computer screen with white text

AI-generated content may be incorrect.

1. Định nghĩa và kết nối các thành phần trong file cấu hình

File logic LightSensorC.nc

* Sử dụng các giao diện đã được kết nối trong LightSensorCApp.nc: Boot (khởi động), Timer0 (định thời), TempRead và LightRead (đọc dữ liệu từ nhiệt độ và ánh sáng)



1. Gọi các biến cần sử dụng

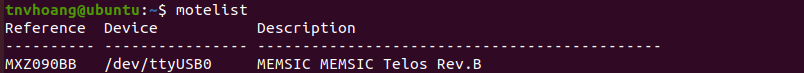
* Khai báo biến celcius lấy giá trị nhiệt độ ở °C, và luminance để lấy giá trị ánh sáng với đơn vị lux
* Bắt đầu bộ định thời lặp lại mỗi 1 giây khi Telos B khởi động Boot.booted()
* Mỗi giây hàm Timer.fired() sẽ yêu cầu đọc dữ liệu từ cảm biến nhiệt độ và ánh sáng
* Khi dữ liệu nhiệt độ đã được đọc xong (TempRead.readDone)
  + Sử dụng result để kiểm tra trạng thái kết quả có thành công chưa
  + Thực hiện chuyển đổi dữ liệu nhiệt độ thô với công thức T(°C) = -39.6 + 0.01 \* val và in ra giá trị đó lên giao diện
* Tương tự, hàm LightRead.readDone cũng xử lí dữ liệu ánh sáng khi cảm biến đã đọc xong
  + Sử dụng công thức luminance = 2.5 \* (val / 4096.0) \* 6250.0 để tính toán giá trị ánh sáng thô sang đơn vị lux, sau đó thực hiện in dữ liệu đó lên giao diện

A screenshot of a computer program

AI-generated content may be incorrect.

1. Viết logic chính cho ứng dụng

## Nạp chương trình



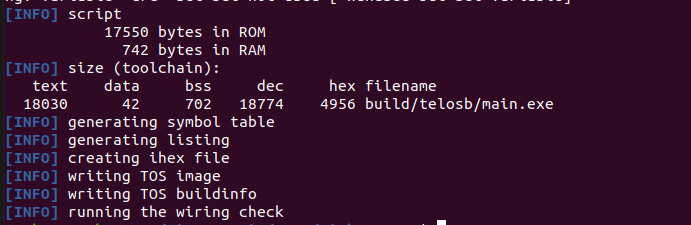
1. Kiểm tra danh sách thiết bị



1. Cấp quyền truy cập vào thiết bị



1. Biên dịch chương trình LightSensor



1. Kết quả biên dịch

A computer screen shot of a program code

AI-generated content may be incorrect.

1. Nạp chương trình vào Telos B

## Kết quả

A screen shot of a computer

AI-generated content may be incorrect.

Video thực hiện câu 2, câu 3:

<https://drive.google.com/file/d/1Y4a7PSebQAqCVZiWRwM9OS5Vowv7U1vu/view?usp=sharing>

**YÊU CẦU CHUNG**

1. Đánh giá

* Chuẩn bị tốt các yêu cầu đặt ra trong bài thực hành.
* Sinh viên hiểu và tự thực hiện được bài thực hành, trả lời đầy đủ các yêu cầu đặt ra.
* Nộp báo cáo kết quả chi tiết những đã thực hiện, quan sát thấy và kèm ảnh chụp màn hình kết quả *(nếu có)*; giải thích cho quan sát *(nếu có)*.
* Sinh viên báo cáo kết quả thực hiện và nộp bài.

1. Báo cáo

* Nộp file .docx. Tập trung vào nội dung, giải thích.
* Nội dung trình bày bằng Font chữ Cambria hoặc Times New Roman (*tuy nhiên, phải chuyển đổi hết báo cáo này sang 1 font chữ thống nhất*) – cỡ chữ 13. Canh đều (Justify) cho văn bản. Canh giữa (Center) cho ảnh chụp.
* Đặt tên theo định dạng: STTNhom-LabX\_MSSV1\_MSSV2. (trong đó X là Thứ tự buổi Thực hành).

Ví dụ: TH1.01-Lab1\_25520001\_25520002

* Nếu báo cáo có nhiều file, nén tất cả file vào file .ZIP với cùng tên file báo cáo.
* Không đặt tên đúng định dạng – yêu cầu, sẽ KHÔNG chấm điểm bài thực hành.
* Nộp file báo cáo trên theo thời gian đã thống nhất tại courses.uit.edu.vn.

Bài sao chép, trễ, … sẽ được xử lý tùy mức độ vi phạm.

**HẾT**