微算機系統實習

實驗三--二

日期:2021/04/09

1. 組員姓名：

第一組

資工二　108590044　何柏憲

資工二　108590452　林峻霆

1. 實驗步驟截圖與說明：

實驗目標:

• 學習如何使用Qt開發嵌入式系統GUI操作界面

• 學習利用Qt操作GPIO上的LED  
• 學習綁定Qt上的快捷鍵  
• 學習設計Qt多個Signal且能夠彼此作動  
• 學習利用Qt Creator完成跨平台編譯開發

實驗過程:

1.創建屬於我們的MainWindow

2.找到我們需要的圖 並一個個的創造我們需要的物件

3.撰寫我們的程式碼（lab2+lab3的內容整合）和增加功能

4.QT Creator 跨平台開發：將建立好的專案 lab03-2 傳到TX2上面

以下是我們的程式碼：

gpio.h

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <errno.h>

#include <unistd.h>

#include <fcntl.h>

#include <iostream>

using namespace std;

int gpio\_export(unsigned int gpio){

int fd,len;

char buff[64];

fd = open("/sys/class/gpio/export",O\_WRONLY);

if (fd<0){

perror("gpio/export");

return fd;

}

len = snprintf(buff,sizeof(buff),"%d",gpio);

write(fd,buff,len);

close(fd);

return 0;

}

int gpio\_unexport(unsigned int gpio){

int fd,len;

char buff[64];

fd = open("/sys/class/gpio/unexport",O\_WRONLY);

if (fd<0){

perror("gpio/unexport");

return fd;

}

len = snprintf(buff,sizeof(buff),"%d",gpio);

write(fd,buff,len);

close(fd);

return 0;

}

int gpio\_set\_dir(unsigned int gpio,string dirStatus){

int fd;

char buff[64];

snprintf(buff,sizeof(buff),"/sys/class/gpio/gpio%d/direction",gpio);

fd = open(buff,O\_WRONLY);

if(fd<0){

perror("gpio/direction");

return fd;

}

if(dirStatus == "out"){

write(fd,"out",4);

}else{

write(fd,"in",3);

}

close(fd);

return 0;

}

int gpio\_set\_value(unsigned int gpio,int value){

int fd;

char buff[64];

snprintf(buff,sizeof(buff),"/sys/class/gpio/gpio%d/value",gpio);

fd=open(buff,O\_WRONLY);

if(fd<0){

perror("gpio/set-value");

return fd;

}

if(value == 0 ){

write(fd,"0",2);

}else{

write(fd,"1",2);

}

close(fd);

return 0;

}

mainwindow.cpp

#include "mainwindow.h"

#include "ui\_mainwindow.h"

#include "gpio.h"

#include <QTimer>

#include <QTime>

bool fuck =true;

int arr\_pin[] = {396,393,255,481};

void turnOn(unsigned int target){

gpio\_export(target);

gpio\_set\_dir(target,"out");

gpio\_set\_value(target,1);

}

void turnOff(unsigned int target){

gpio\_set\_value(target,0);

gpio\_unexport(target);

}

void delay( int millisecondsToWait )

{

QTime dieTime = QTime::currentTime().addMSecs( millisecondsToWait );

while( QTime::currentTime() < dieTime )

{

QCoreApplication::processEvents( QEventLoop::AllEvents, 100 );

}

}

MainWindow::MainWindow(QWidget \*parent) :

QMainWindow(parent),

ui(new Ui::MainWindow)

{

ui->setupUi(this);

}

MainWindow::~MainWindow()

{

delete ui;

}

void led\_choose(unsigned int x , bool t){

if(t){

gpio\_export(x);

gpio\_set\_dir(x,"out");

gpio\_set\_value(x,1);

}

else{

gpio\_set\_value(x,0);

gpio\_unexport(x);

}

}

void MainWindow::shine(QLabel \*l){

l->setHidden(false);

}

void MainWindow::off(QLabel \*l){

l->setHidden(true);

}

void MainWindow::check(QCheckBox \*c, bool t){

c->setChecked(t);

}

void MainWindow::on\_shining\_clicked()

{

ui->led1->setHidden(!ui->check\_led1->isChecked());

ui->led2->setHidden(!ui->check\_led1\_2->isChecked());

ui->led3->setHidden(!ui->check\_led1\_3->isChecked());

ui->led4->setHidden(!ui->check\_led1\_4->isChecked());

led\_choose(arr\_pin[0],ui->check\_led1->isChecked());

led\_choose(arr\_pin[1],ui->check\_led1\_2->isChecked());

led\_choose(arr\_pin[2],ui->check\_led1\_3->isChecked());

led\_choose(arr\_pin[3],ui->check\_led1\_4->isChecked());

}

void MainWindow::on\_Switching\_on\_clicked()

{

fuck = true;

QTimer \*timer = new QTimer(this);

connect(timer, SIGNAL(timeout()), this, SLOT(update()));

timer->start(1000);

int t = ui->spinBox->value();

t+=1;

int i =0 ;

while(fuck)

{

if(i % 2 ==0 ){

shine(ui->led1);

shine(ui->led2);

off(ui->led3);

off(ui->led4);

check(ui->check\_led1,true);

check(ui->check\_led1\_2,true);

check(ui->check\_led1\_3,false);

check(ui->check\_led1\_4,false);

turnOn(arr\_pin[0]);

turnOn(arr\_pin[1]);

turnOff(arr\_pin[2]);

turnOff(arr\_pin[3]);

delay(5000/t);

}

else{

shine(ui->led3);

shine(ui->led4);

off(ui->led1);

off(ui->led2);

check(ui->check\_led1\_3,true);

check(ui->check\_led1\_4,true);

check(ui->check\_led1\_2,false);

check(ui->check\_led1,false);

turnOff(arr\_pin[0]);

turnOff(arr\_pin[1]);

turnOn(arr\_pin[2]);

turnOn(arr\_pin[3]);

delay(5000/t);

}

i++;

}

}

void MainWindow::on\_Switching\_off\_clicked()

{

turnOff(arr\_pin[0]);

turnOff(arr\_pin[1]);

turnOff(arr\_pin[2]);

turnOff(arr\_pin[3]);

fuck=false;

off(ui->led3);

off(ui->led4);

off(ui->led1);

off(ui->led2);

check(ui->check\_led1\_3,false);

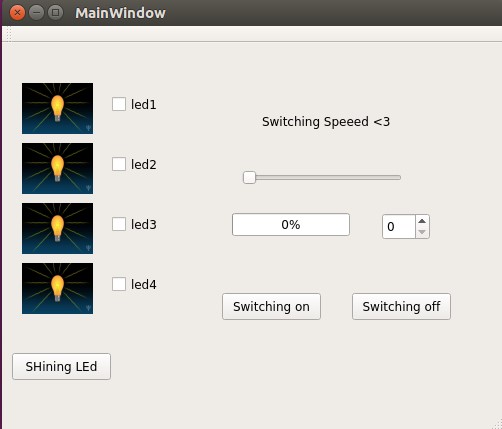
check(ui->check\_led1\_4,false);

check(ui->check\_led1\_2,false);

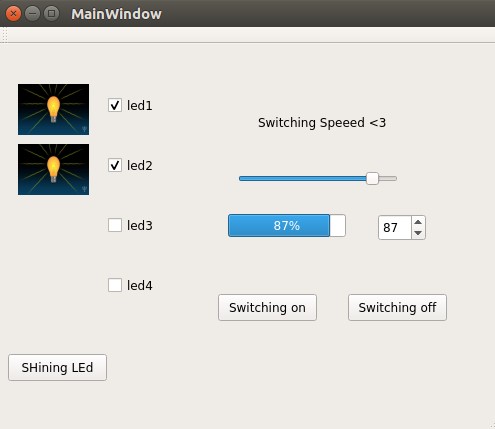
check(ui->check\_led1,false);

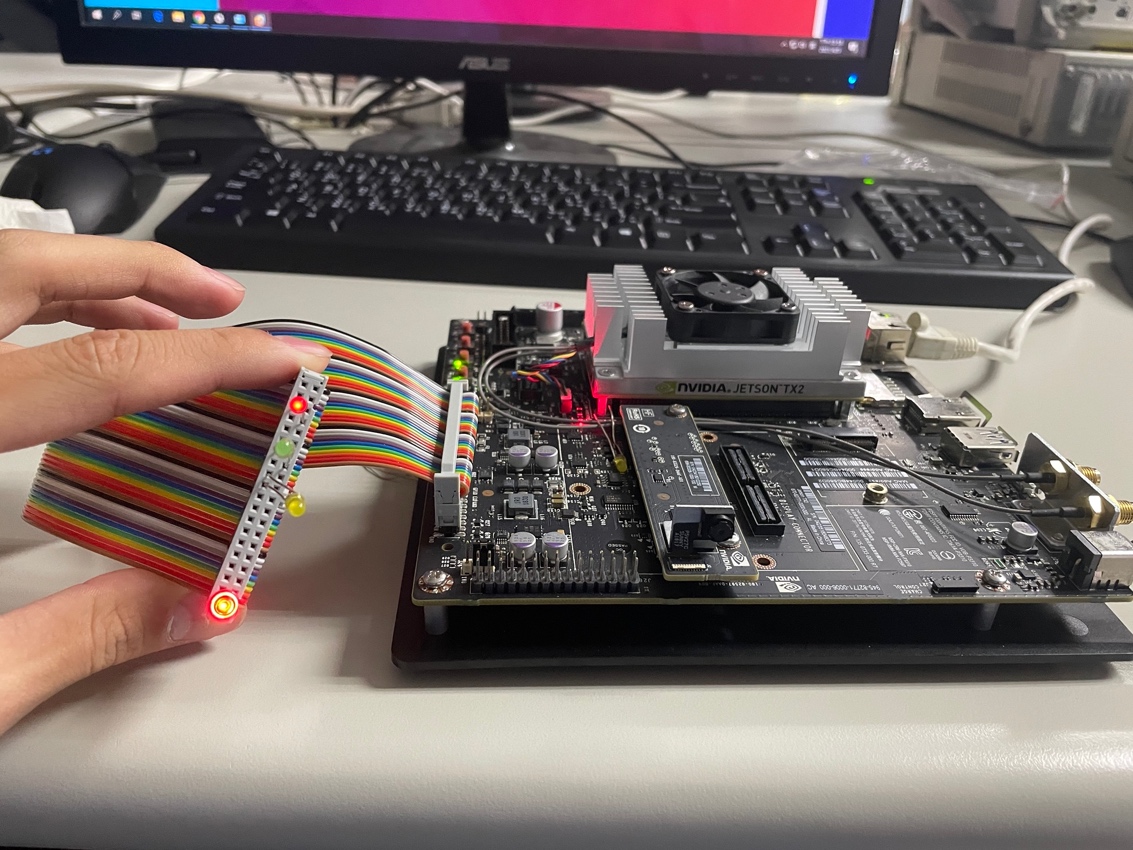
}

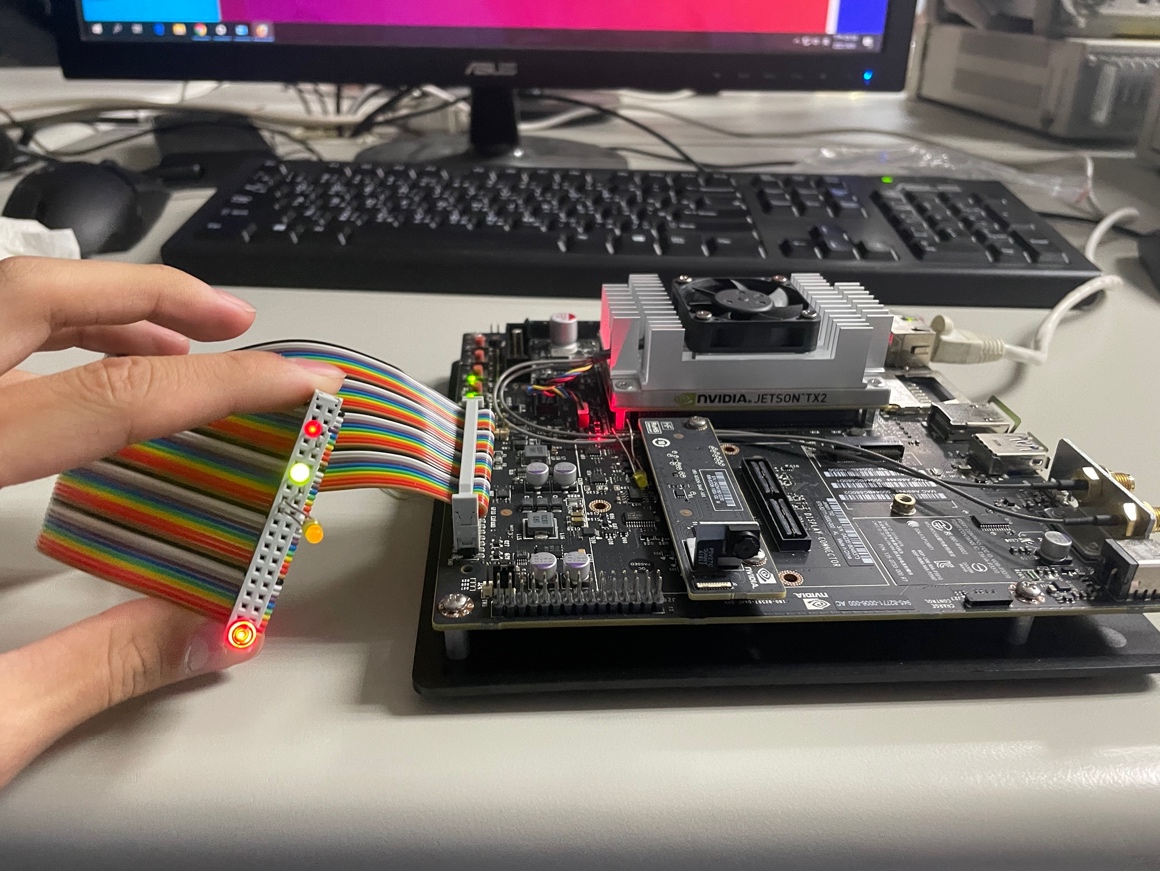
以下是實驗結果



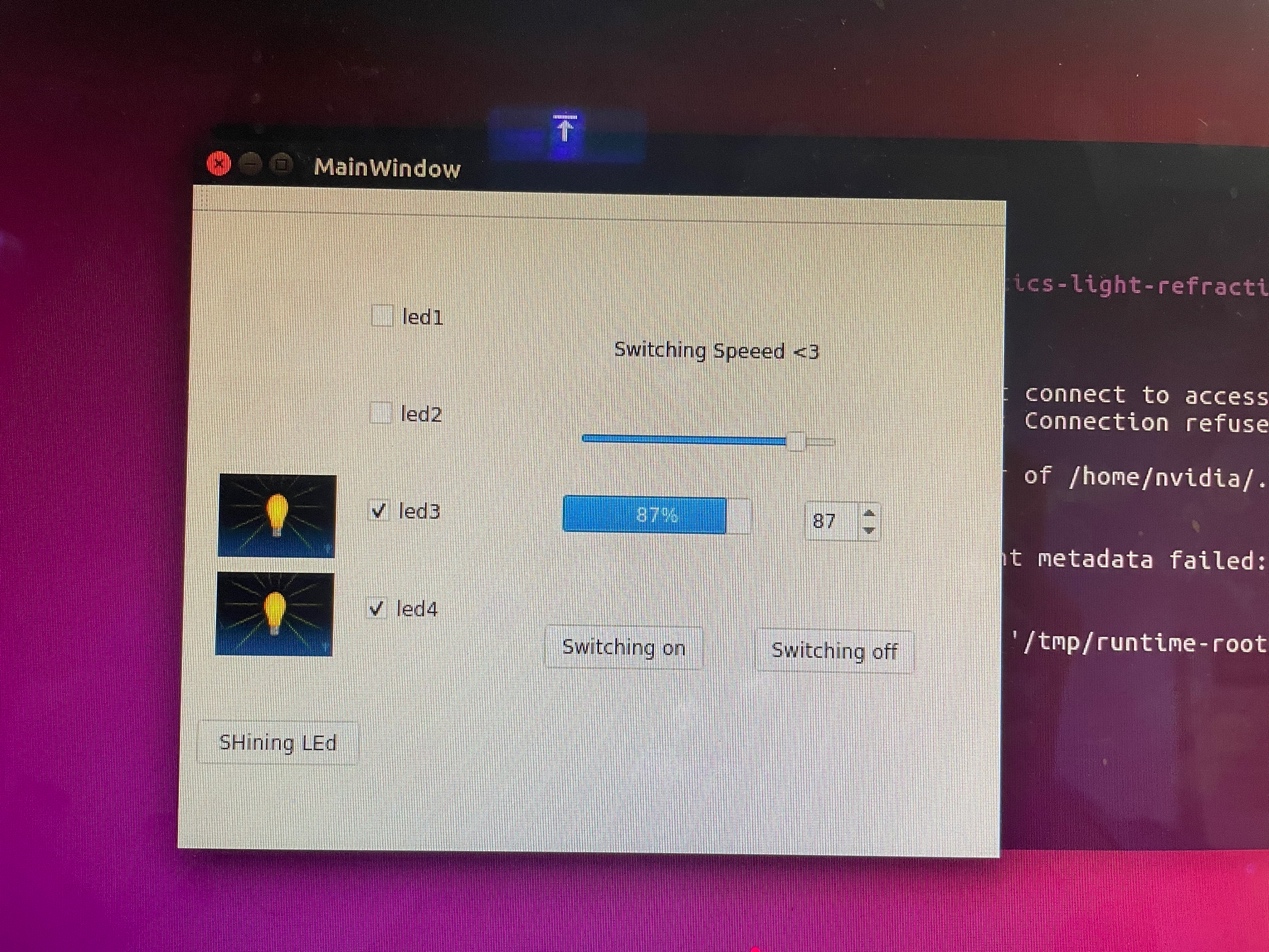
MainWindow

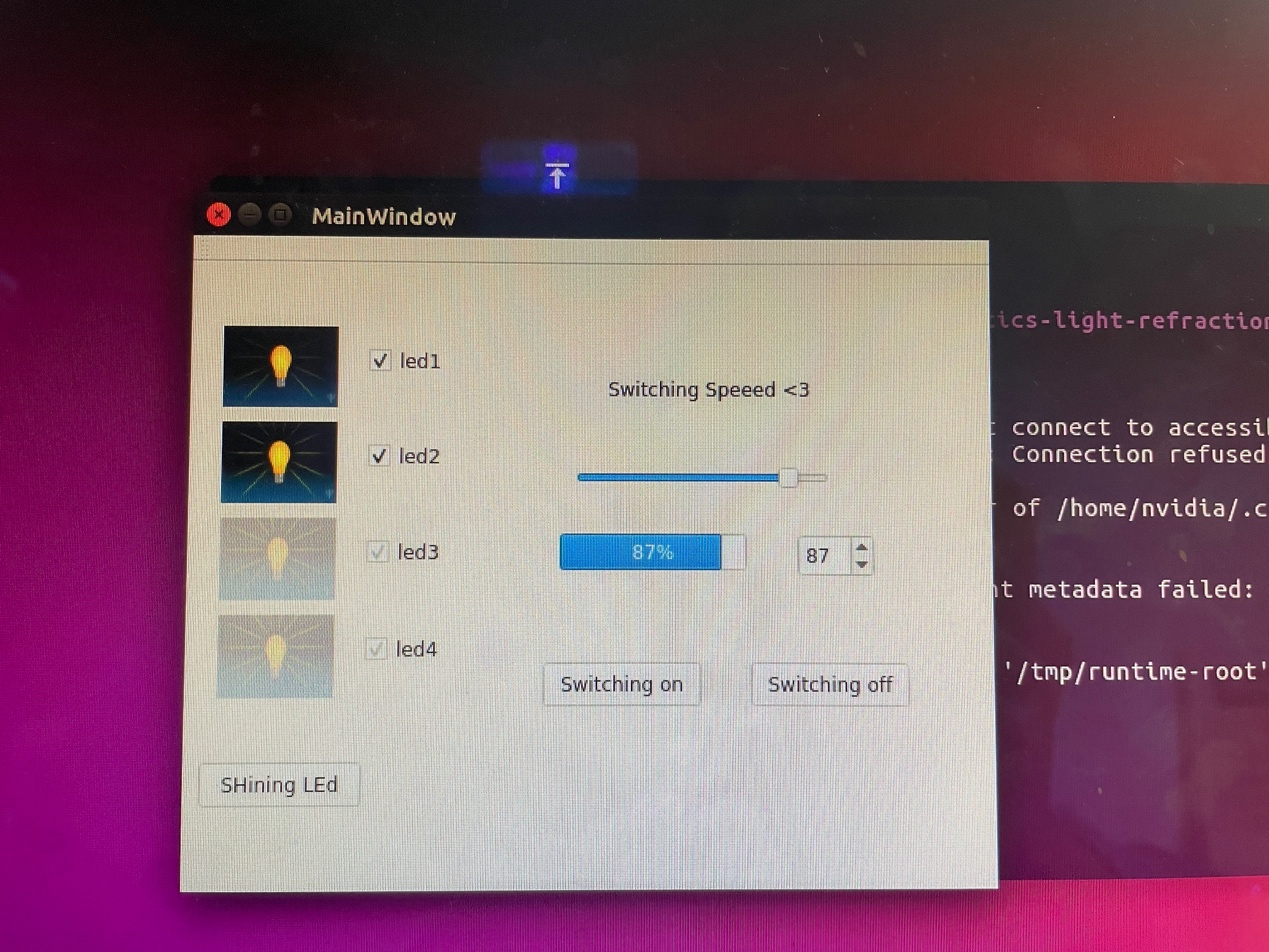
設定led1 led2 為 87%的速度 正準備交替閃爍





設定led1 led2 為 87%的速度 交替閃爍





因為速度太快拍照時產生了殘影ＸＤ

影片連結：

1. 組員貢獻比例：

何柏憲：50%

林峻霆：50%

1. 心得：

何柏憲：

林峻霆：