

Voici le programme du coté de visual C++ :ca
fonctionne ! sous cette ecriture !

```
#include "CArdCom.h"
#include "stdafx.h"

int _tmain(int argc, _TCHAR* argv[])
{
    CArdCom TEST;

    //startting communication
    if(!TEST.Init("COM4",115200,8,NOPARITY,ONESTOPBIT)) printf("error
opening\n");
    else printf("opening ok\n");

    if(!TEST.SetPin(22)) printf("light");
    else TEST.SetPin(22);
    if(!TEST.SetPin(2)) printf("light");
    else TEST.SetPin(2);
    //if(!TEST.ResetPin(2)) printf("Pas light\n");
    //else TEST.ResetPin(2);

    char message[50];
    if(!TEST.SendMessageArd(message)) printf("not sent or NCK\n");

    system("pause");
    return 0;
}
```

**Et voici le programme dans Arduino ; la
seule partie modifié afin que je puisse
commandé de visual C++ est ecrit en rouge
en contre bas !**

Nota :

digitalWrite(pin,1); le 1=hight et 0= low

```
/*ARDUINO - communication
 *Start - 25.6.2009
 */

#define START 0x01
#define END 0x02
#define STOP 0x03

#define ADR //adress of the first LED pin
```

```

class Communication
{
    private:
        int numberBytes;           //number of all received bytes
        int received;              //flag for receiving message
        int processed;             //flag for processing message
        char message[200];         // message
        char iddef[4];             //identificator
        char body[20];
        void ReceiveMessage();     //read message from serial port
        void ProcessMessage();     //process message
        void GetIddef();           //get IDEF of message
        int GetBody();             //get body of message
        void Set(int);             //set target pin to 1
        void Reset(int);           //reset target pin to 0
        void DoLight();            //
        int brightness;
        int pin;
    public:
        Communication();           //costructor
        void Init();               //inicialization
        void Send(char*);          //send a char to serial port
        void Send(int);            //send int
        void Send(float);          //send float
        void HandlMessage();       //handl message from PC
        ~Communication();          //destructor
};

int StrToInt(char *str, int count)
{
    int i = 0;
    int number = 0;

    for(i=count; i>=0; i--)
    {
        number += (str[i] - 48) * round(pow(10, count - i));
        //strlen(str)
    }
    return number;
}

Communication::~~Communication()
{
}

/*-----*/
Communication::Communication()
{
    numberBytes = 0;
    received = 0;
    processed = 0;
}

/*-----*/

```

```

void Communication::Init()
{
    Serial.begin(115200);
}
/*-----*/
void Communication::Set(int pin)
{
    digitalWrite(pin, 1);
    pinMode(pin, OUTPUT);
}

/*-----*/
void Communication::Reset(int pin)
{
    digitalWrite(pin, 0);
    pinMode(pin, OUTPUT); //ici eteint la lumiere
}

/*-----*/
int Communication:: GetBody()
{
    int i = 0;

    while((message[i+3] != END)&&(i<20))
    {
        body[i] = message[i+3];
        i++;
    }

    //body[i] = 0x00;

    return StrToInt(body, i-1);
}

/*-----*/
void Communication::Send(char *info)
/*form of message:
*  START|IDEF|message|END |SUM|*****STOP
*  0x01 |XXX |body..|0x02|X |*****0x03
*/
{
    Serial.print(START, BYTE);    //Start
    Serial.print(info);           //ACK, NCK, message
    Serial.print(END, BYTE);      //End
    Serial.print(3, DEC);         //sum
    Serial.print(STOP, BYTE);     //STOP
}

/*-----*/
void Communication::Send(int info)
{
    Serial.print(START, BYTE);    //Start
    Serial.print("GET");
}

```

```

        Serial.print(info, DEC);          //value
        Serial.print(END, BYTE);          //End
        Serial.print(3, DEC);             //sum
        Serial.print(STOP, BYTE);         //STOP
    }

/*-----*/
void Communication::Send(float info)
{
    Serial.print(START, BYTE);            //Start
    Serial.print("MES");
    Serial.print(info, DEC);              //value
    Serial.print(END, BYTE);              //End
    Serial.print(3, DEC);                 //sum
    Serial.print(STOP, BYTE);             //STOP
}

/*-----*/
void Communication::GetIdef()
{
    int i = 0;
    for(i=0; i<3; i++)idef[i] = message[i];
    idef[3] = 0x00;
}

/*-----*/
void Communication:: DoLight()
{
    int i = 0;
    char LED[100];

    while((message[i+3] != END))
    {
        LED[i] = message[i+3];
        i++;
    }

    //to do add code for writting value to PWM pin, the values are stored in LED
    array
    //values 0x00, 0x01, 0x02, 0x03 are reservad for the communication and cannot
    by used

}

/*-----*/
void Communication::ReceiveMessage()
{
    char Cpom;

    do
    {
        Cpom = Serial.read();
        switch (Cpom)
        {
            case START:{
                received = 0;
                numberBytes = 0;
            }
        }
    }
}

```

```

        }break;
    case END:{
        message[numberBytes] = END;
        numberBytes++;
    }break;
    case STOP:{
        message[numberBytes] = STOP;
        received = 1;
    }break;
    default:{
        message[numberBytes] = Cpom;
        numberBytes++;
    }
}
} //switch
}while ((Serial.available() > 0));
}

/*-----*/
void Communication::HandleMessage()
{
    if(Serial.available() > 0) ReceiveMessage();
    if(received == 1) ProcessMessage();
}

/*-----*/
void Communication::ProcessMessage()
{
    received = 0; //clear flag

    GetIdef();
    if(strcmp("CON",idef) == 0)
    {
        processed = 1;
        Send("ACK");
        //set flag
    }
    if(strcmp("LIG",idef) == 0)
    {
        DoLight();
        processed = 1;
        Send("ACK");
        //set flag
    }
    if(strcmp("SET",idef) == 0)
    {
        Set(GetBody());
        processed = 1;
        Send("ACK");
        //set flag
    }
    if(strcmp("RES",idef) == 0)
    {
        Reset(GetBody());
        processed = 1;
        Send("ACK");
        //set flag
    }
}

```

```

    if(!processed)
        Send("NCK");                                //send error

    processed = 0;                                    //clear flag
}
/*-----*/
/*-----*/
/*-----*/

Communication Com;

/*-----*/
void setup()
{
    Com.Init();
}

/*-----*/
void loop()
{
    Com.HandleMessage();
}

```

Mais voila je ne trouve pas le moyen de pouvoir varier l'intensité de la led de 0 à 255. Comme je pouvais le réaliser auparavant avec mon ancien programme (que j'envoyé simplement sur la carte avec le logiciel arduino.)

```

int brightness;
int pin;
void setup()
{}

void loop()
{
    {pin=2;    int brightness =11;
    analogWrite(pin, brightness);}

    {pin=22;    digitalWrite(pin,1;
    pinMode(pin, OUTPUT);}
    {pin=23;    digitalWrite(pin,0;
    pinMode(pin, OUTPUT);}
}

```