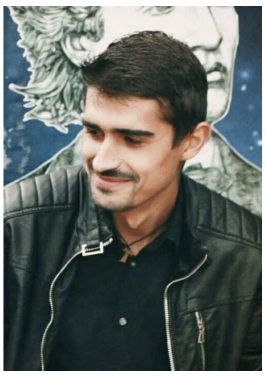
Powered Lamp Dimmer using XMC2GO



Ignat Daniel-Andrei, 1B **Pascaru Daniela, 1A**

danielignat13@gmail.com danielapascarum@gmail.com

More science, less programming

1. Summary

We often adjust the display brightness of our mobile phones to suit to our need. With this project, you can do that for your bedside lamps or any other lighting at home. We are going to teach you how to make a lamp dimmer. Using this project, you can control the brightness of your table lamp according to your needs and start building one!

2. Introduction

This system is useful for using almost at everything in order to control even the brightness of a lamp, or just the intensity of an electronic device. This kind of idea can also be implemented in a computer cluster, in order to perform some predefined tasks, controlled and scheduled by a software.

In this project, we are going to adjust the brightness of the lamp connected to the circuit using a bluetooth module. The brightness can be changed according to the commands we provide to the serial port. We will be using these particular commands in this project:

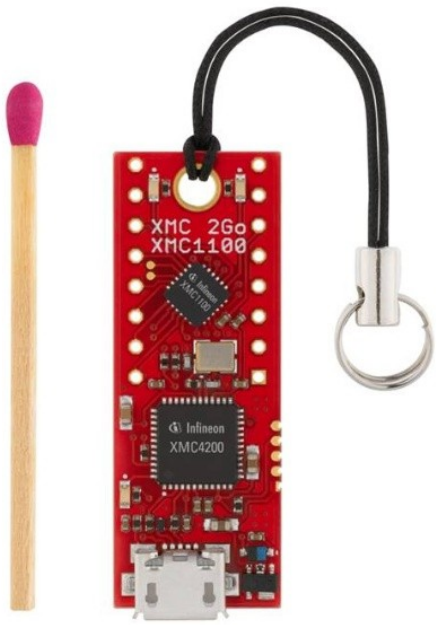
- ‘0’ to TURN OFF.
- ‘1’ for 20% brightness.
- ‘2’ for 40% brightness.
- ‘3’ for 60% brightness.
- ‘4’ for 80% brightness.
- ‘5’ for 100% brightness.

Required Materials:

- XMC2GO
- HC-05 Bluetooth Module
- Resistors
- Optocoupler(4N35)
- Diodes
- Capacitors
- MOSFET(IRF830A)
- Lamp: 100W
- 230V Supply

- Socket
- Solder dot board and Soldering kit

XMC2GO:

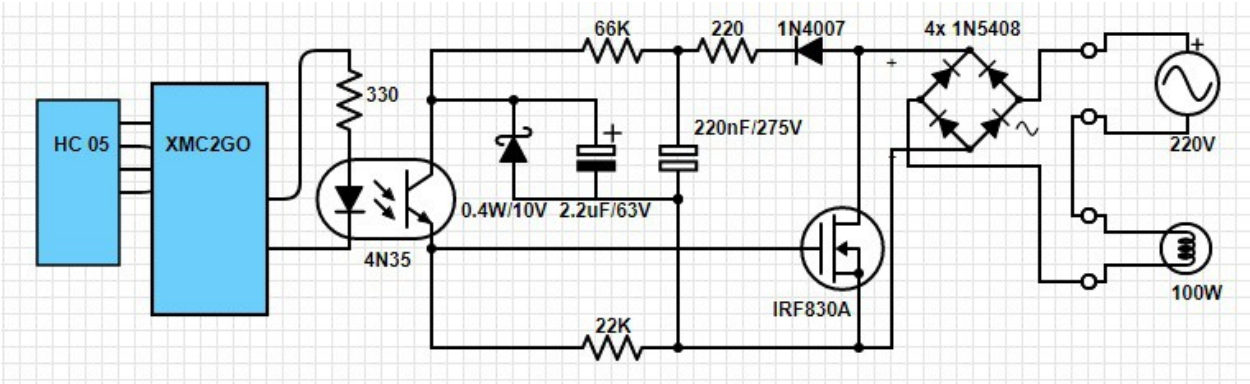


Summary of Features:

- XMC1100 (ARM® Cortex™-M0 based)
- On-board J-Link Lite Debugger
- Power over USB (Micro USB)
- ESD and reverse current protection
- 2 x user LED
- Pin Header 2x8 Pins suitable for Breadbord

3. Solution description, implementation details

We designed a Pulse Width Modulated (PWM) dimmer circuit which uses an IRF830A in a diode bridge which is used to control the voltage across the light bulb with pulse wave modulation (PWM). The power supply voltage for driving the gate is supplied with the voltage across the Metal Oxide Semiconductor Field-Effect Transistor (MOSFET).

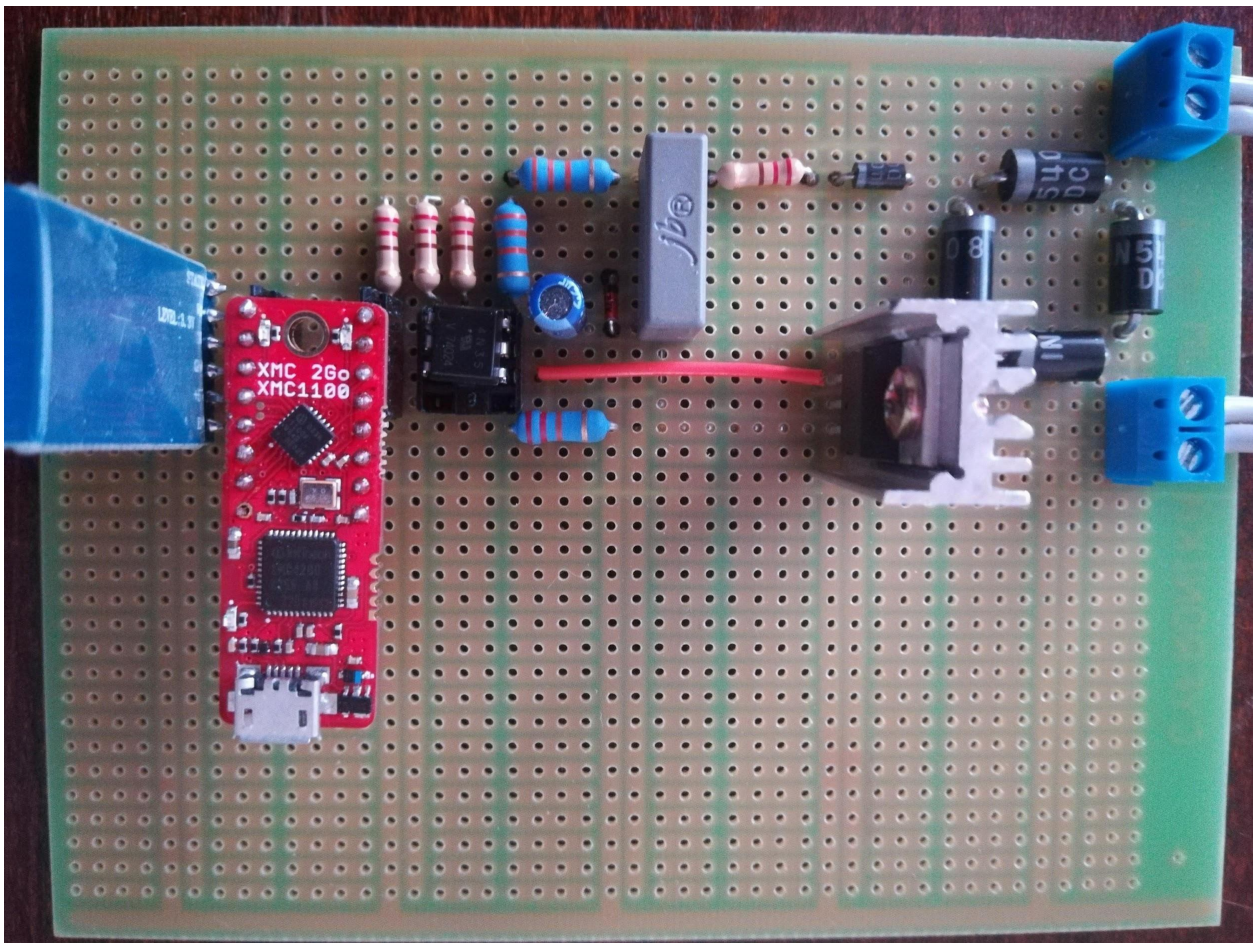


The figure above explains the positioning of the different electrical components in the circuit. You can follow the circuit diagram to solder your dot board.

The rectifier bridge is used to convert from AC to DC.

The optocoupler(4N35) is used for driving the gate. The function of the 330 resistor is to protect the LED in the optocoupler. Like 220, 22K resistors also limits the current so that a 'hard' voltage can be applied safely. The optocoupler provides class-II isolation. (A class-II or double insulated electrical appliance is the one which has been designed in such a way that it does not require a safety connection to electrical earth). This is good enough to ensure safety to the regulator.

Now we need the conduction in the MOSFET as quickly as possible. For this, we connect the transistor in the optocoupler to the positive power supply. The MOSFET is switched asynchronously and this can cause the DC current to flow. A low pass filter is used to maintain the voltage constant.



For programming we used DAvE that is a digital Application Virtual Engineer and uses C/C++ software development and code generation tool for microcontroller applications. DAvE is a standalone system with automatic code generation modules and is suited to develop software drivers for Infineon microcontrollers and aids the developer with automatically created C-level templates and user desired functionalities.

Source code:

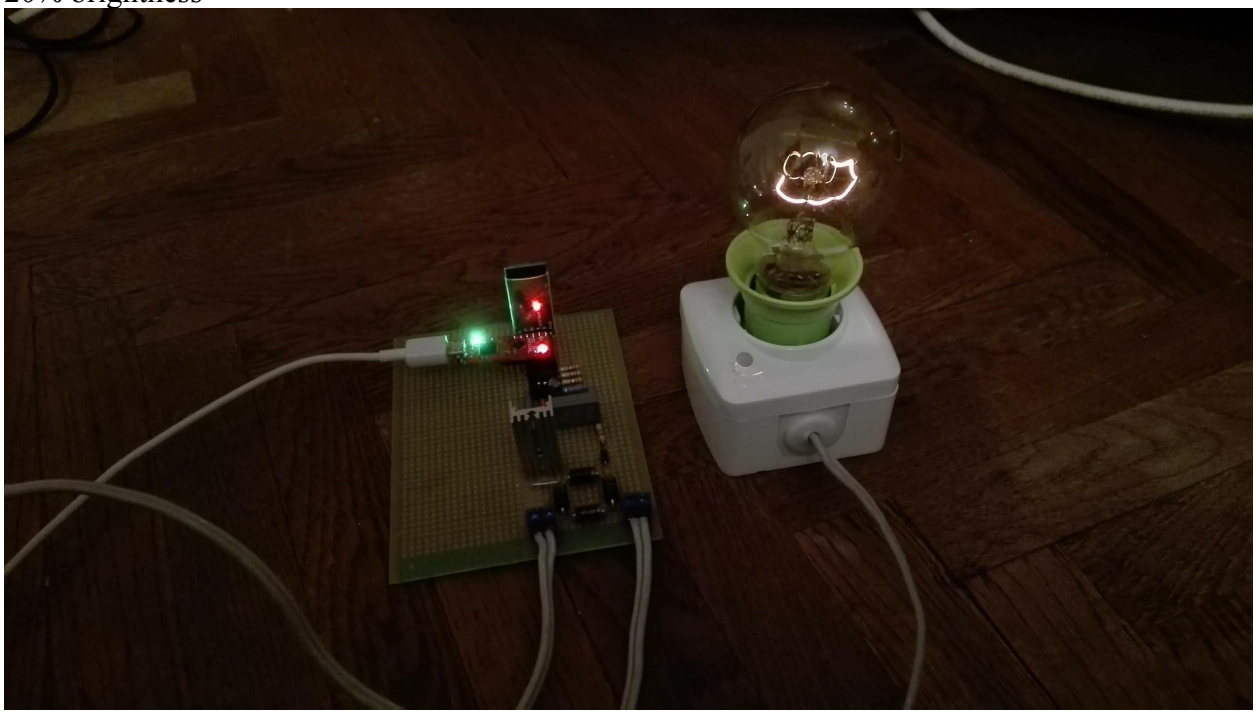
```
#include <DAVE.h>
int main(void)
{

   uint8_t j;
   DAVE_Init();
   UART_SetRXFIFOTriggerLimit(&UART_0, 0);
   PWM_Init (&PWM_0);         // PWM Init
   PWM_SetFreq(&PWM_0,50);     // Set PWM freq
         PWM_SetDutyCycle(&PWM_0,0); // Set DutyC 0
         PWM_Start(&PWM_0);          // Start PWM

  while(1)
  {
     if(UART_GetRXFIFOStatus(&UART_0))
```

```
        {
            UART_ClearRXFIFOStatus(&UART_0, 1);
            j = UART_GetReceivedWord(&UART_0);
            UART_TransmitWord(&UART_0,j);
        if(j=='0')
        {
          PWM_SetDutyCycle(&PWM_0,0);
        }
        else if(j=='1')
        {
          PWM_SetDutyCycle(&PWM_0,2000);
        }
        else if(j=='2')
        {
          PWM_SetDutyCycle(&PWM_0,4000);
        }
        else if(j=='3')
        {
          PWM_SetDutyCycle(&PWM_0,6000);
        }
        else if(j=='4')
        {
          PWM_SetDutyCycle(&PWM_0,8000);
        }
        else if(j=='5')
        {
          PWM_SetDutyCycle(&PWM_0,9999);
        }
        }
  }
  return 0;
}
```

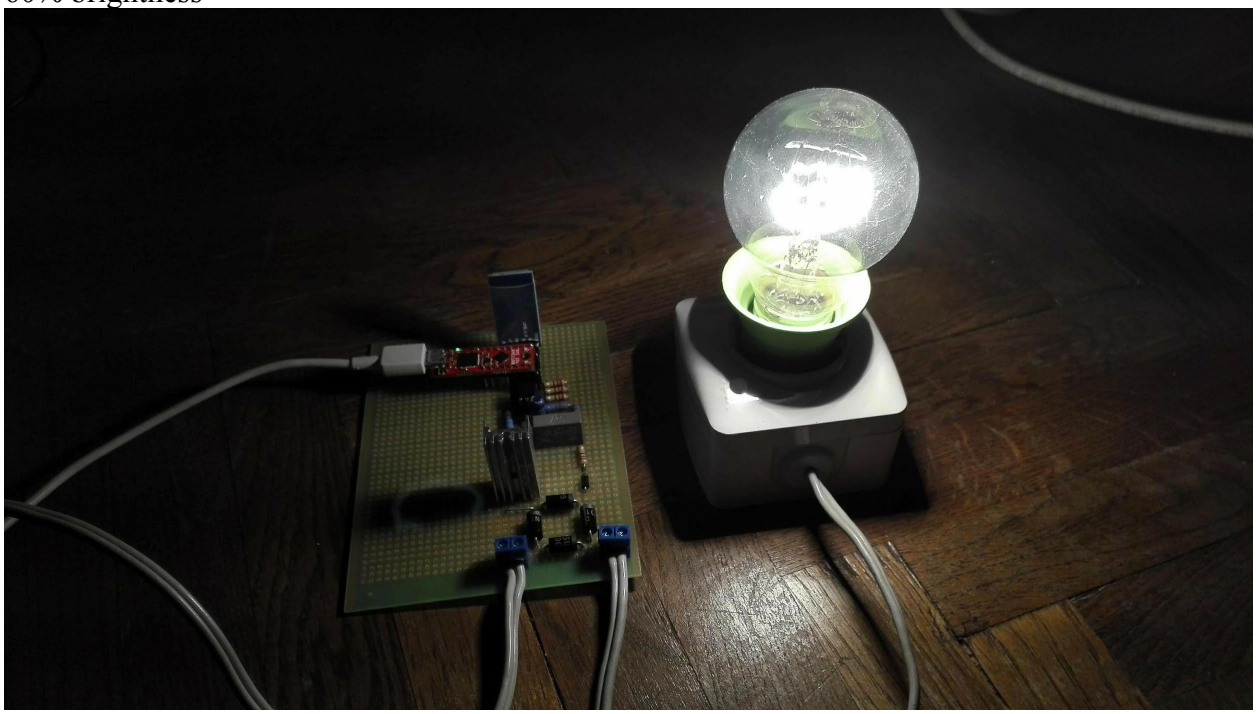
4. Results presentation

The following pictures present the working system:

20% brightness



60% brightness



100% brightness

5. Bibliography

- http://www.alldatasheet.com/datasheet-pdf/pdf/26143/VISHAY/4N35.html
- http://www.alldatasheet.com/datasheet-pdf/pdf/250861/VISHAY/IRF830A.html
- https://www.infineon.com/cms/en/product/evaluation-boards/kit_xmc_2go_xmc1100_v1/
- https://www.infineon.com/dgdl/Board_Users_Manual_XMC_2Go_Kit_with_XMC1100_R1.0.pdf?fileId=db3a3043444ee5dc014453d6c75078c6
- http://embedac.ro/CI/index.html