



INSTITUTO SUPERIOR TÉCNICO

UC: Sistemas de Controlo Distribuído em Tempo-Real

Controlo em Tempo-Real de um Sistema de Iluminação

Grupo nº 31

João Morais, nº 83916

Francisco Freiria, nº 97236

10 de janeiro de 2020

Índice

1. Introdução	3
2. Sistema de medição de iluminância.....	4
2.1 Não-Linearidade do sistema	4
2.2 Calibração.....	5
2.3 Estado estacionário.....	6
2.4 Step response.....	7
2.5 Controlo local	8
2.6 Controlo Feedback e Feedforward	9
3. Controlo global.....	10
3.1 Sistema de controlo não-cooperativo	10
3.2 Sistema de Calibração	11
3.3 Comunicação CAN-BUS	12
3.4 Métricas de performance	13
4. Código	14
4.1 Informações adicionais	14
4.2 Tabela de comandos	16
5. Conclusão	18

1. Introdução

No âmbito da Unidade Curricular de Sistemas de Controlo Distribuídos em Tempo-Real foi-nos proposta a realização de um projeto com a finalidade de implementar um sistema de controlo em tempo real para um sistema de iluminação, com o intuito de aplicar os conhecimentos adquiridos ao longo do semestre relativamente a sistemas de controlo, programação de microcontroladores, métodos de troca de informação, entre outros.

Este projeto apresenta-se com um protótipo de um escritório com várias luminárias e surge na ideia que cada vez mais se procura variadas soluções eficientes para a redução do consumo energético assegurando sempre o conforto aos utilizadores da mesma. O objetivo será manter, em cada secretaria e dependendo da sua ou não ocupação, uma quantidade fixa de luz, escolhida pelo utilizador.

Assim é apresentada uma solução para controlar dois LED's a partir de leituras lidas num LDR. O processamento de toda a informação fica a cargo de dois arduinos com um microcontrolador ATmega328p [1], um para cada secretária do escritório e a troca de informação a processar a cargo de dois MCP2515 [2] sendo este uma comunicação CAN-BUS.

Será ainda implementado um conjunto de funções que permitem obter várias informações, tanto relativas ao funcionamento, estados de ocupação, Lux lidos, etc. como também métricas de performance como por exemplo o *visibility error* e o *flicker error*. Será ainda possível alterar níveis de luminosidade pretendidos ou ocupação da secretaria para que possam ser os utilizadores da mesma a alterar as condições de conforto do escritório.

2. Sistema de medição de iluminância

2.1 Não-Linearidade do sistema

Para a sala ter a quantidade de luz desejada é necessário o auxílio de uma fonte de luz, e tendo em conta que o nosso sistema é uma mera demonstração recorreu-se a um led. Para este ter a luz mais apropriada a um menor consumo energético, mas sem nunca provocar aos utilizadores da mesma uma quantidade de luz inadequada, é necessário fornecer uma determinada quantidade de tensão. Para tal tivemos de estudar o comportamento do pin A0 (Analog Pin) do arduino face a um sinal analógico, ou na verdade PWM [3].

Como o PWM é um conceito que serve apenas para simular um sinal analógico através do pulsar de um sinal digital então também é necessário compreender como este se comporta.

Para melhor interpretar os valores fornecido no nosso pin A0 testamos para todos os valores de PWM a resposta do mesmo. O gráfico obtido na Figura 1 representa o comportamento do PWM face á tensão no pin A0 e, por conseguinte, a resistência do LDR conclui-se que não varia de uma forma linear, ou seja, evolui não-linearmente com o aumento da tensão. Para tal foi tido em consideração a tensão máxima fornecida pelo arduino de 5V, concluímos que 5V correspondem a 255 bits de PWM. Isto deve se ao facto do nosso microcontrolador processar a informação a 8 bits.

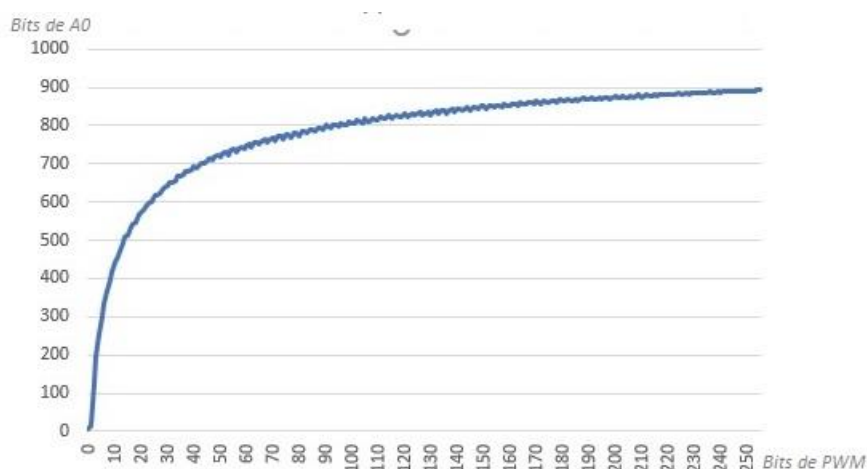


Figura 1:Variação dos valores lidos no pino A0 com o aumento do valor de PWM

2.2 Calibração

Segundo o datasheet do LDR GL5528 [4], a resistência (kΩ) que produz em relação á luminância (LUX) varia linearmente numa escala logarítmica (Figura 2). A zona a sombreado da Figura 2 representa o intervalo de valores válidos.

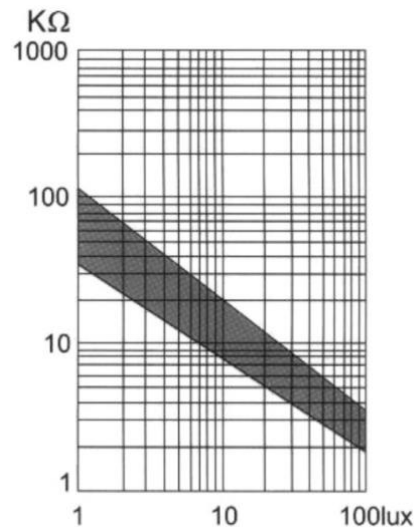


Figura 2: Relação entre a Resistência e a Luminosidade [4]

Assim sendo foi necessário definir a equação que modelaria os valores reconhecidos pelo LDR. Equação esta também fornecida pelo datasheet do GL5528 [4] e apresentada na seguinte forma:

$$\log_{10} R_{LDR} = m \log_{10} L + b \quad (1)$$

Resolvendo a equação substituindo por valores na faixa de valores válidos dados pelo datasheet do LDR, foram calculados os valores que correspondem ao declive da reta m e b . Os valores obtidos encontram-se na Tabela 1.

	LDR
m	-0.7797
b	4.8603

Tabela 1: Valores calculados para modelar os valores lidos do LDR

Apresentando então a equação da seguinte forma:

$$\log_{10} R_{LDR} = -0.7797 \log_{10} L + 4.8603 \quad (2)$$

Tendo em consideração todos os valores que a função pode tomar concluímos que a equação (2) se enquadra dentro de todos os valores fornecidos pelo gráfico apresentado no datasheet.

Para os valores lidos pelo LDR, no pin A0, tivemos em consideração que sendo esta uma entrada de ADC e que o ADC interno do arduino funciona internamente com 10bits, foi feita uma conversão dos valores recebidos de 5V para 1024, isto é 2^{10} .

2.3 Estado estacionário

A fim de testar a resposta no estado estacionário do sistema de luminância, foram testados, para todos os valores de PWM, de 0 a 255, os valores da luminância correspondente. Para tal recorremos à equação (2). Obtivemos as respostas de cada impulso a cada 100 milissegundos para obter valores mais estáveis. Leituras num intervalo de tempo mais curto não seriam tão fidedignas uma vez que os valores então apresentados seriam antes de uma estabilização do sistema.

A partir do gráfico da Figura 3 concluímos que, ao incrementar a tensão no led (ou seja no PWM), esta evolui linearmente com a leitura do LDR, em Lux. Em suma, a relação do aumento da tensão no led face aos LUX's lidos pelo LDR e por se apresentarem de uma forma aproximadamente linear, apresentam um ganho DC constante.

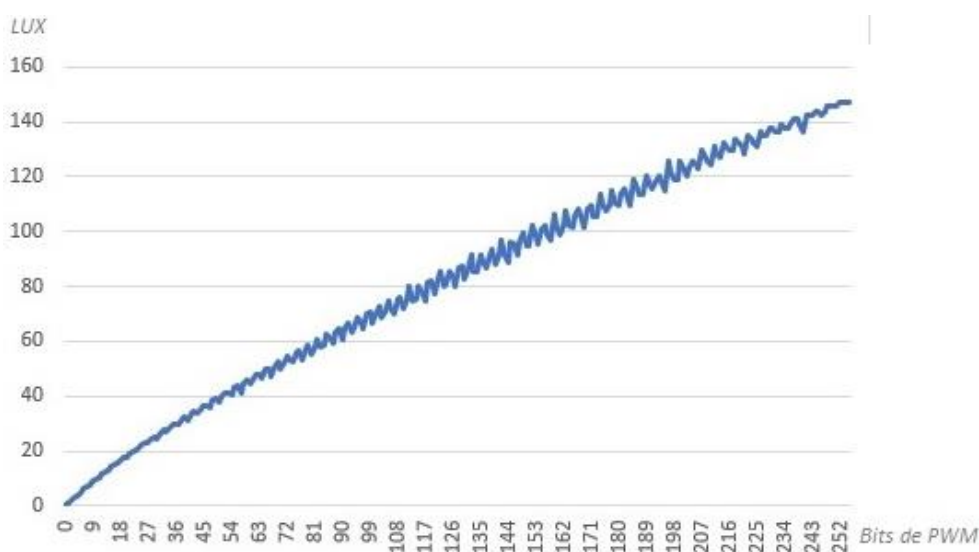


Figura 3: Valores lidos no pin A0 em LUX face ao aumento de PWM

O sistema comporta-se não linearmente quanto à tensão recebida no pin A0 mas linearmente face aos LUX que essa tensão representa.

2.4 Step response

Quando fornecido ao sistema um valor de luminância conhecido ao LDR, como vimos anteriormente, o sistema reage como um sistema linear de primeira ordem, pois o polinómio é de primeira ordem. Desta forma, podemos resumir o seu comportamento à equação (3). Assim, é também possível determinar um ganho K_0 , linear, dado pela relação de saída e de entrada do sistema num estado estacionário. Para tal esta relação tem de ser feita face à linearidade do sistema, entre LUX e Volts.

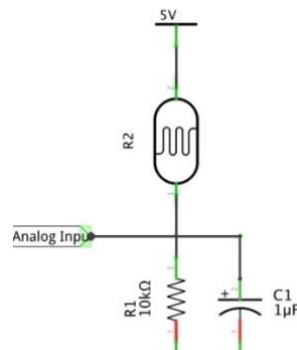


Figura 4: Circuito do LDR

$$\dot{v}\tau(x) = -v + V_{cc} \frac{R_1}{R_1 + R_2(x)} \quad (3)$$

com:

$$\tau(x) = R_{eq}(x)C_1 \quad (4)$$

$$R_{eq}(x) = \frac{R_1 R_2(x)}{R_1 + R_2(x)} \quad (5)$$

Foi também considerada uma contante de tempo, τ , que corresponde ao intervalo de tempo que o sistema demora a atingir um estado de estabilidade de leitura dos valores.

2.5 Controlo local

Após a análise do comportamento do LDR bem como do funcionamento do sistema, foram definidos valores de referência, em LUX, para o estado de ocupação e de não ocupação de cada secretaria. Na Tabela 2 são apresentados os estados de ocupação para cada um dos LDR's. Após uma análise dos valores recebidos por cada LDR e tendo em conta os Lux's obtidos na segunda parte do projeto, concluímos que para esta fase, o estado de ocupação e não ocupação seriam os definidos para o LDR1 pois para este seria impossível a deteção de 70 LUX.

O controlo local aplicado recebe como entradas o valor de referência de iluminância desejada e o valor lido pelo LDR, os dois em LUX. O valor lido pelo LDR é convertido em Lux pois, como visto anteriormente, a relação Luminância-Tensão é uma relação linear. Este controlo tem como output o valor de PWM (de 0 a 255 bits) a aplicar na fonte luz. Foram ainda implementados um controlo de feedback e um controlo de feedforward que permitem o ajuste da luz emitida pelo LED, isto é, o microcontrolador vai comparar sempre o valor lido no LDR com o valor de referência e ajustar o valor de PWM a cada ciclo. Assim, sempre que necessário, o valor de PWM, é ajustado com mais 1 ou menos 1 bit a cada ciclo. O tempo de cada ajuste é de aproximadamente 25 milissegundos, tempo este que depende do número de instruções executadas pelo microcontrolador no loop principal do código. Cada ajuste (aumento ou diminuição de 1 bit de PWM) representa um aumento de aproximadamente 0.02 V no led.

Na primeira fase do projeto, o controlo local aplicado apresenta-se diferente do aplicado na fase final do mesmo. Primeiramente planeou-se que o controlo devia ser feito com o valor absoluto da diferença da luminância de referência face á luminância lida pelo LDR. Esta diferença, representando uma quantidade de luminância, poderia ser convertida em PWM e subtraída ou somada aos bits que tinham sido enviado para o LED na iteração anterior, dependendo se fosse preciso diminuir ou aumentar, respetivamente, a luminância lida. Uma vez que os resultados apresentavam-se mais satisfatórios com o ajuste de mais ou menos 1 bit a cada ciclo acabamos por implementar este método na fase final do projeto.

Com a implementação do controlo de feedback e feedforward foram ainda elaboradas margens de controlo, isto é, sempre que for necessária mais luz que do que o led pode enviar, consequentemente mais tensão que pode ser enviada pelo PWM, o sistema atinge um estado de saturação não enviando mais tensão do que a que o próprio pode enviar. Ainda que a luminância não esteja nos valores de referência, o sistema não permite que o pin do PWM do microcontrolador entre num estado de overflow, evitando por consequência flickers exacerbados.

	<i>Ocupado</i>	<i>Não ocupado</i>
LDR1	20 LUX	10 LUX
LDR2	70 LUX	10 LUX
LDR1+LDR2	20 LUX	10 LUX

Tabela 2: LUX desejados

2.6 Controlo Feedback e Feedforward

Como referido anteriormente, este controlo é feito tendo em base o valor de referência da luminância da sala face ao valor lido, e que, dependendo se o valor é superior ou inferior à referência, ajusta a quantidade de PWM a enviar ao led.

Observando a Figura 5, onde é apresentado o gráfico que mostra a resposta deste sistema de controlo, conclui-se que o método aplicado permite uma resposta bastante satisfatória.

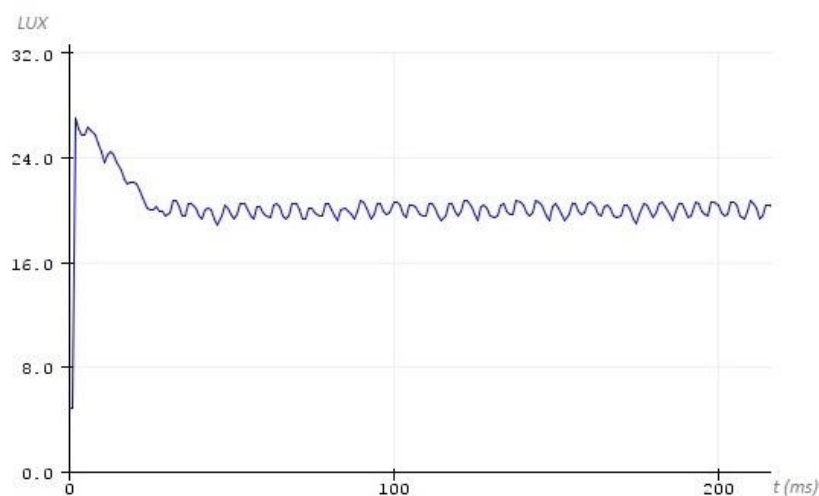


Figura 5: Desempenho do controlador para uma referência de 20 lux.

Na Figura 6 é apresentado o gráfico que mostra a resposta do sistema a uma perturbação externa. Esta perturbação corresponde a uma luz exterior adicionada ao sistema (primeiro pico) e depois retirada (segundo pico). É possível observar que o sistema responde bastante rápido à alteração de luminosidade, ajustando bastante rapidamente o valor aos Lux's desejados.

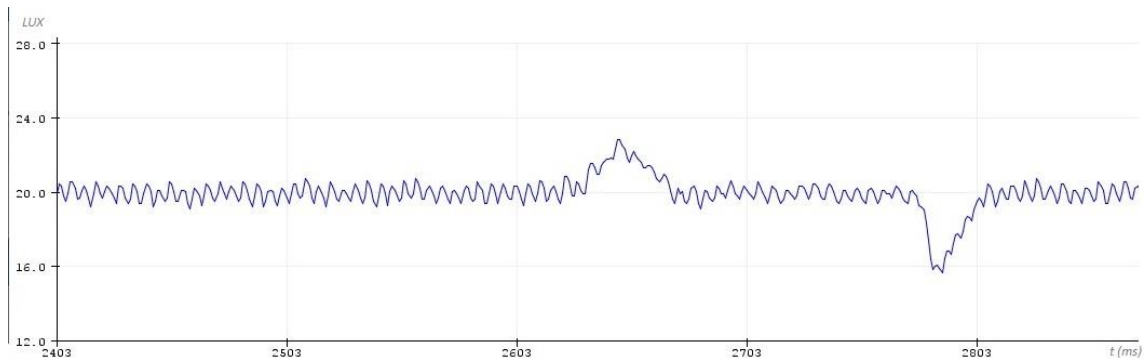


Figura 6: Resposta do controlador a uma perturbação externa, para uma referência de 20 lux.

Sendo o nosso sistema um sistema variável no tempo, produz um erro a ele associado. O erro das medições face ao valor de referência de luminância para o protótipo é de aproximadamente 1,5 LUX.

3. Controlo global

3.1 Sistema de controlo não-cooperativo

Quando os dois sistemas de controlo local são colocados numa área comum, um controlo local, torna-se insuficiente. A luz emitida por um led vai influenciar a leitura do LDR da outra mesa. Para além disso, luz externa também influencia as leituras feitas pelo LDR. Então para que todos estes fatores fossem considerados e de modo a obtermos o menor consumo energético possível, elaborou-se um sistema de controlo global, isto é, sempre que são retirados os valores de LUX's do LDR, para o cálculo do valor de PWM a enviar para o próprio led, é considerado a influência da luz externa e a do outro LED.

Assim o sistema implementado que permite o controlo do PWM foi um sistema de controlo não cooperativo, isto é, um sistema que ainda que tenha em consideração fatores externos, apenas se preocupa em achar o seu próprio objetivo. Os fatores externos são considerados, armazenando os valores de cada mesa em diferentes variáveis para poderem ser usados futuramente no programa. Este sistema tem como desvantagens um grande esforço computacional e o facto de apenas se guardarem os valores de cada iteração nessa mesma iteração.

3.2 Sistema de Calibração

Uma vez que o este projeto é apenas um protótipo e devido ao facto de ter de se transportar, todos os ruídos elétricos, condições de luz, etc., vão ser sempre diferentes. Então, para que toda esta entropia nas medições do LDR fosse reduzida foi necessário recorrer ao desenvolvimento de um sistema de calibração para o nosso sistema, sempre que o código fosse inicializado. Para tal deve-se proceder ao reset manual dos dois arduinos em simultâneo. Numa situação de maior estabilidade, esta calibração deverá ser feita aquando a manutenção e verificação do equipamento.

O sistema de calibração é utilizado para calcular os ganhos na relação tensão-luminância. Este ganho refere-se, no nosso caso, a quantos LUX vão ser detetados a mais entre o caso em que são enviados 0bits e o caso em que são enviados 100bits para o LED.

Tendo em conta que o nosso sistema possui dois LED's, para além do próprio ganho, é necessário ter ainda em conta a influência que a luz de cada LED tem no outro LDR. O cálculo dos ganhos individuais e dos "ganhos cruzados" está representado nas equações que se seguem:

$$k_{11} = \frac{l_1}{V_1}; k_{22} = \frac{l_2}{V_2}; k_{12} = \frac{l_1}{V_2}; k_{21} = \frac{l_2}{V_1} \quad (6)$$

A Figura 7 ilustra o nosso sistema quanto à influência que cada led tem em cada LDR.

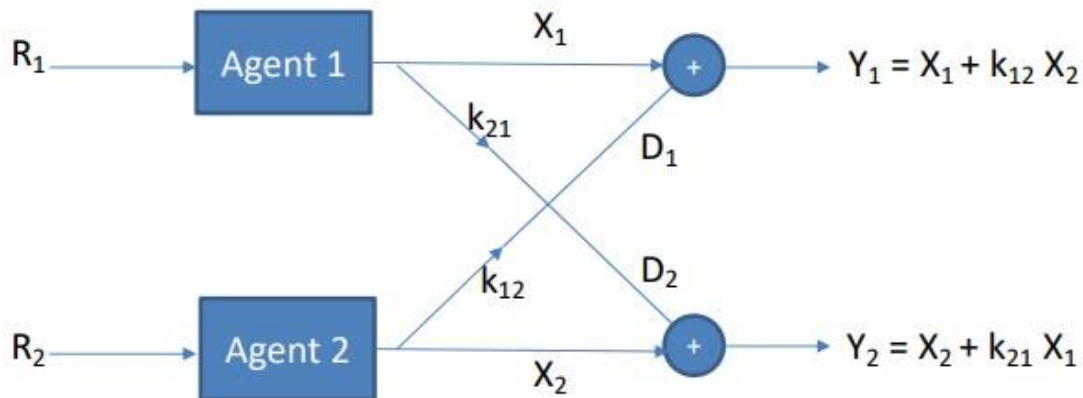


Figura 7: Representação dos led's e da influência que os mesmo têm nos LDR's [5]

Foi tido em consideração que cada cálculo do ganho só deve ser efetuado após a estabilização do LED no valor pretendido e por isso, entre cada leitura do valor do LDR, há um delay com o valor da constante de tempo de estabilização do LED para que os valores sejam os mais próximos do correto.

3.3 Comunicação CAN-BUS

A fim de obtermos um menor consumo energético, e sabendo que o comportamento de cada LED influencia o comportamento do outro LED foi então necessário o utilizar o recurso a um método de comunicação. O método utilizado é via CAN-BUS que utiliza como protocolo de comunicação SPI [5]. O dispositivo utilizado que conecta cada nó é o MCP2515.

Foram então definidas duas funções no código implementado que permitem o envio e receção de dados, "Read" e "Write". O tempo médio para que uma mensagem fosse recebida é muito baixo pois esta informação é enviada/recebida em cada ciclo do loop, sendo os valores guardados em variáveis e usados futuramente, se necessário. Estes tempos variam, pois, dependem sempre do tamanho da mensagem a enviar/receber.

3.4 Métricas de performance

Para avaliar a performance do controlo e para garantir o menor consumo energético bem como a maior confortabilidade aos utilizadores da sala, foram definidas algumas métricas de performance. Elas são então, o consumo energético (7), o erro de visibilidade (8) e o erro dos flickers (9), sendo as últimas duas utilizadas como critério de conforto. Em todas elas o objetivo foca-se em ser o menor possível. O consumo energético interessava ser o menor possível para que o investimento no controlo da sala tivesse retorno na redução da eletricidade gasta. O erro de visibilidade é uma boa métrica de performance pois permite avaliar, ao longo do tempo, se a confortabilidade das condições de luz está a ser corretamente aplicada. Já o erro de flicker é outra métrica de avaliação importante pois permite a avaliar se a variação de luminância pode ser pejorativa aos utilizadores da sala.

As fórmulas que permitem o cálculo das mesmas são:

$$E_j = P_j \sum_{i=1}^N d_{i-1}(t_i - t_{i-1}) \quad (7)$$

$$V = \frac{1}{N} \sum_{i=1}^N \max(0, L(t_i) - l(t_i)) \quad (8)$$

$$F = \frac{1}{N} \sum_{i=1}^N f_i \quad (9)$$

COM:

$$f_i = \begin{cases} (|l_i - l_{i-1}| + |l_{i-1} - l_{i-2}|) / (2T_s) & \text{if } (l_i - l_{i-1}) \times (l_{i-1} - l_{i-2}) < 0 \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

4. Código

4.1 Implementação

O programa começa com a inicialização das variáveis. As variáveis usadas são bastantes pois os resultados dos cálculos de ambas as secretarias são guardados no arduino em que se está a trabalhar.

A seguir é apresentada a função `pedido()`. Esta função é chamada dentro do loop para verificar se existem mensagens a ler na janela de comandos e tem como funcionalidade dar resposta aos diferentes comandos apresentados na Tabela 3 ou “err” se a mensagem não for compatível com nenhuma das apresentadas.

O `Setup()` começa e aqui é adquirido o valor do “decisionPin”. Este pin está ligado a HIGH num dos arduinos e a LOW no outro e permite fazer a distinção do arduino em que queremos trabalhar, dentro do programa. Depois são calculados os ganhos dos dois LDR’s. Estes ganhos serão depois utilizados na determinação inicial do melhor valor de bits a enviar a cada led de modo a obter os LUX’s desejados. Inicialmente, e com os dois leds desligados, determina a luz exterior presente na caixa (em Lux) em ambas as secretarias. Depois, liga um dos leds com 100bits, mantendo o outro desligado, determinando aqui a influencia da luz do led nele próprio e a influencia que este tem no outro LDR. Este último processo é depois repetido para o outro led. Os ganhos são depois calculados a partir da equação (11).

$$GANHO = ((Lux_lido(100bits) - Lux_lido(0bits)) / (100bits - 0bits)) \quad (11)$$

Após estes cálculos, o programa faz a decisão dos LUX’s desejados em cada secretaria, dependendo da sua ocupação. Os valores escolhidos foram 20Lux no caso de a secretaria estar ocupada e 10 Lux no caso contrário. Depois, e dividindo os LUX’s desejados pelo ganho, é determinado o valor de bits a enviar inicialmente ao led de

modo a aproximá-lo do valor pretendido, apesar de não ter em conta a luz proveniente do outro led, ou a exterior.

Depois desta atribuição, começa o `loop()` do programa. Aqui, inicialmente, é feita a leitura dos bits recebidos nos LDR's de ambos os circuitos e esta informação enviada para o outro arduino. A partir desta leitura, é determinado se a quantidade de luz é maior ou menor do que a pretendida, fazendo atualizações constantes do valor de bits a enviar ao led, subtraindo um bit se o valor lido for maior do que o pretendido, ou somando caso contrário. Caso o valor de bits pretendido seja maior ou menor do que o possível para o led, o valor é atualizado para 255 ou para 0, respetivamente. Este método funciona como um método não cooperativo pois cada led é atualizado apenas a partir do valor lido na “sua” secretaria. Quando tentámos aplicar um controlo cooperativo estávamos a obter resultados mais insatisfatórios pois ocorriam muitos *flickers*. Apesar disso, os resultados com este processo são bastante positivos.

Depois deste ajuste, é obtido o tempo decorrido nessa iteração (necessária ao cálculo da energia (8)) e atualizado o tempo decorrido desde o início do programa. Após esta leitura é calculada a energia consumida e potência e atualizados os valores da energia e potência totais.

De seguida é feito o cálculo e atualização do *visibility error* de ambas as secretarias. Depois de fazer a comparação entre os LUX's lidos e os desejados e no caso de os lidos serem menores, faz o cálculo do erro a partir da equação (8) e atualiza os erros de ambos e o total.

No fim do `loop()` é feito o cálculo do *Flicker error*. Aqui são guardados os valores de LUX's lidos em três ciclos consecutivos. No fim do terceiro ciclo a condição (10) é posta em prática com os valores adquiridos e é feito o cálculo do *Flicker error* de cada uma das secretarias, através da equação (9). Estes valores são depois somados para o cálculo do *Flicker error* total.

4.2 Tabela de comandos

Comando	Pedido do Cliente	Resposta Programa	Observações
Obter a iluminância no momento na secretaria <i>.	"g l <i>"	"l <i> <val>"	<val> em Lux
Obter o Duty Cycle no momento no led <i>.	"g d <i>"	"d <i> <val>"	<val> em percentagem
Obter a ocupação da secretaria <i> no momento.	"g o <i>"	"o <i> <val>"	<val> ocupado -> 1 desocupado -> 0
Alterar a ocupação na secretaria <i>.	"o <i> <val>"	"ack" ou "err"	Apenas é possível fazer na secretaria em que estamos a trabalhar.
Obter a patamar mínimo de iluminância para o estado ocupado na secretaria <i>.	"g O <i>"	"O <i> <val>"	<val> em Lux
Alterar o patamar mínimo de iluminância para o estado ocupado na secretaria <i>.	"O <i> <val>"	"ack" ou "err"	Apenas é possível fazer na secretaria em que estamos a trabalhar(decision==0). É também apenas possível escolher para <val> três valores: 20, 30 e 40 Lux.
Obter a patamar mínimo de iluminância para estado desocupado na secretaria <i>.	"g U <i>"	"U <i> <val>"	<val> em Lux.
Alterar o patamar mínimo de iluminância para o estado desocupado na secretaria <i>.	"U <i> <val>"	"ack" ou "err"	Apenas é possível fazer na secretaria em que estamos a trabalhar(decision==0). É também apenas possível escolher para <val> três valores: 7, 10 e 15 Lux.
Obter o patamar minimo de iluminancia na secretaria <i>.	"g L <i>"	"L <i> <val>"	<val> em Lux.
Obter a iluminancia externa na secretaria <i>	"g x <i>"	"x <i> <val>"	<val> em Lux.

<i>Obter a referência de controlo da iluminancia na secretaria <i>.</i>	<i>"g r <i>"</i>	<i>"r <i> <val>"</i>	<i><val> em Lux.</i>
<i>Obter o custo da energia na secretaria <i>.</i>	<i>"g c <i>"</i>	<i>"c <i> <val>"</i>	<i><val> em euros.</i>
<i>Obter o consumo de potência instantâneo na secretaria <i>.</i>	<i>"g p <i>"</i>	<i>"p <i> <val>"</i>	<i><val> em Watt.</i>
<i>Obter o consumo de potência instantâneo no sistema.</i>	<i>"g p T"</i>	<i>"p T <val>"</i>	<i><val> em Watt.</i>
<i>Obter o tempo decorrido desde o ultimo reset.</i>	<i>"g t <i>"</i>	<i>"t <i> <val>"</i>	<i><val> em segundos.</i>
<i>Obter o consumo de energia acumulado na secretaria <i>.</i>	<i>"g e <i>"</i>	<i>"e <i> <val>"</i>	<i><val> em Joule.</i>
<i>Obter o consumo total de energia acumulado.</i>	<i>"g e T"</i>	<i>"e T <val>"</i>	<i><val> em Joule.</i>
<i>Obter o visibility error acumulado na secretaria <i>.</i>	<i>"g v <i>"</i>	<i>"v <i> <val>"</i>	<i><val> em Lux.</i>
<i>Obter o visibility error total do sistema.</i>	<i>"g v T"</i>	<i>"v T <val>"</i>	<i><val> em Lux.</i>
<i>Obter o flicker error acumulado na secretaria <i>.</i>	<i>"g f <i>"</i>	<i>"f <i> <val>"</i>	<i><val> em Lux/seg.</i>
<i>Obter o flicker error total do sistema.</i>	<i>"g f T"</i>	<i>"f T <val>"</i>	<i><val> em Lux/seg.</i>
<i>Obter os ganhos do Sistema</i>	<i>"ganhos"</i>		<i>Ganhos dos LDR's face aos LED's em Lux/V</i>

Tabela 3: Tabela de comandos

5. Conclusão

Findo o trabalho podemos concluir que o objetivo principal do mesmo foi cumprido: um baixo custo energético garantindo as condições de conforto aos utilizadores do escritório.

Enfrentamos bastantes adversidades, desde a aplicação do controlo local às comunicações feitas entre os microcontroladores. Os principais problemas na realização do projeto dizem respeito ao controlo cooperativo, que não foi aplicado. Porém a alternativa utilizada apresenta bons resultados, uma vez que são obtidos resultados bastante satisfatórios quanto às métricas de performance e quanto às leituras lidas. Outro fator que foi adverso foi o facto dos LDR usados apresentarem ganhos tão diferentes, restringindo as condições de conforto da sala.

Ficou ainda por aplicar a interface em C++ que permitia um maior conforto na execução da tabela de comandos. No nosso caso foi implementada através do serial monitor do arduino.

Faltou também conseguir alterar as condições de confortabilidade na secretaria do arduino em que não estamos a trabalhar pois houve complicações na forma como a comunicação para este efeito seria feita.

Na generalidade, com a realização do trabalho e considerando todas as adversidades a ele inerentes, podemos concluir que foi bastante enriquecedor em termos de conteúdos aprendidos e que, de uma forma geral, foi possível por em pratica os conteúdos lecionados na cadeira e obtidos resultados bastante satisfatórios.

Referências

[1] ATmega328p

http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf

[2] MCP2515

<http://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf>

[3] PWM

<https://www.arduino.cc/en/tutorial/PWM>

[4] LDR GL5528

<https://pi.gate.ac.uk/pages/airpi-files/PD0001.pdf>

[5] Slides das aulas teóricas

<https://fenix.tecnico.ulisboa.pt/downloadFile/845043405496924/MODULE19-DISTR-NON-COOP-CONTROL.pdf>