

SmartCap

Capacete de segurança automatizado

Thiago Gomes de Sousa Bezerra
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
thiagotnd@hotmail.com

Diogo Gomes de Sousa Bezerra
Universidade de Brasília
Faculdade Gama
Gama, Distrito Federal
diogogsb@hotmail.com

I. INTRODUÇÃO

Na sociedade contemporânea é comum a prática de exploração em minas, as quais tem por objetivo a extração de determinados minérios, como é o caso do carvão e das pedras preciosas. Tal prática é uma das várias responsáveis pelo crescimento da economia mundial, entretanto, os profissionais que trabalham nesses locais podem estar sujeitos a determinados perigos e condições adversas, perigos os quais podem, até mesmo, leva-los a óbito.

Dentre essas ameaças, existem aquelas que são, muitas das vezes, imperceptíveis para os trabalhadores, como é o caso dos vazamentos de gases nocivos à saúde, tais como CO, H₂S, NO, NO₂ e O₂, produzidos pela utilização de explosivos e motores a combustão[4], além do gás natural. Estes são responsáveis por explosões, desmaio de trabalhadores em zonas de difícil acesso e riscos futuros a saúde. Pode-se citar também a exposição a altas temperaturas, que podem causar doenças e/ou agravantes das mesmas, como é o caso de queimaduras de pele.

Além do mais, é comum situações que impossibilitam a prática do trabalho em questão, como é o caso da falta de iluminação em determinados pontos dessas minas, a qual pode ser ocasionada por diversas situações.

II. OBJETIVO

Através de um capacete de segurança, conseguir identificar situações adversas provenientes do meio, como cavernas ou/ambientes de mineração, tais como a identificação de gases tóxicos prejudiciais a saúde e falta de luminosidade. Assim, quando ocorrer tais identificações o dispositivo enviará ao usuário um alerta de segurança.

Com tudo, o intuito deste projeto é a automação de um equipamento de segurança individual (EPI), mais especificamente o capacete de segurança, com o cuidado de minimizar e prevenir situações adversas ao usuário.

III. BENEFÍCIOS E REQUISITOS

A partir das situações apresentadas, propõem-se o desenvolvimento de um equipamento autônomo para suprir tais necessidades, o SmartCap.

O SmartCap será um equipamento que auxiliará os trabalhadores em determinadas ocasiões. Este auxílio será feito através da comunicação entre um microcontrolador (MSP430) e sensores capazes de captar diversas perturbações, como a presença de gases nocivos e a falta de luminosidade, além de

componentes que irão auxiliar e avisar. Quando o sensor de gás nocivo identificar a presença desses gases, o equipamento retornará uma resposta ao usuário através de um sinal sonoro, proveniente de um buzzer. Já quando o sensor de luminosidade (LDR) identificar a falta de luminosidade no ambiente, uma lanterna fixada ao capacete será ligada, afim de iluminar o ambiente.

IV. SOFTWARE E HARDWARE

MATERIAIS UTILIZADOS

Na tabela 1 pode-se observar os materiais utilizados para o desenvolvimento do projeto.

Tabela 1 – Materiais utilizados no desenvolvimento do projeto.

Materiais utilizados
Capacete de Segurança
MSP430
Sensor de gás – MQ2
Sensor de luminosidade – LDR
Jumpers
Protoboard
Led
Buzzer
Resistor ()
Potenciômetro 1kΩ

HARDWARE

O LDR (Light Dependent Resistor) é basicamente um sensor fotossensível cuja resistência varia conforme a incidência de luz sobre seu corpo, este sensor é mostrado na figura 1.1.[6] Este sensor é composto por uma área exposta à luz feita de material semicondutor que varia sua resistência elétrica conforme o nível de incidência de raios luminosos; isto porque todo material semicondutor é sensível à radiação de alguma forma (luminosa ou não, visível ou não). A resistência desse componente diminui ao aumentar a incidência de luz no mesmo, como mostrado no gráfico presente na figura 1.2.



Figura 1.1: LDR

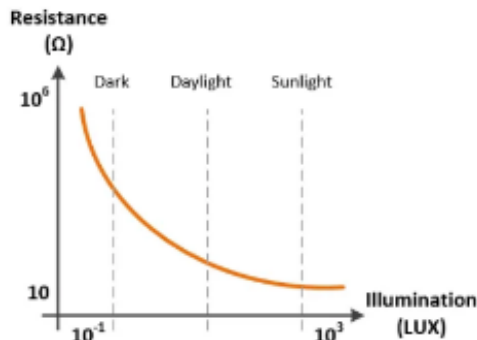


Figura 1.2: Variação da resistência conforme a incidência de luz.

O sensor de gás utilizado para o projeto é o MQ-2, [5] este sensor é capaz de detectar concentrações de gases combustíveis e fumaça no ar. Quando a concentração de gases fica acima do nível, o qual é ajustado por um potenciômetro presente no módulo utilizado, a saída digital D0 fica em estado alto, caso contrário a saída fica em estado baixo. Para ter uma resolução melhor e medir a variação da concentração dos gases no ar é possível usar a saída analógica A0 e conectar a um conversor AD (analógico/digital), como o presente no microcontrolador MSP430 utilizado por exemplo. O MQ-2 é capaz de detectar GLP, Metano, Propano, Butano, Hidrogênio, Álcool, Gás Natural, além de outros gases inflamáveis. Na figura 2 pode-se observar o sensor de gás MQ-2 a ser utilizado no projeto.

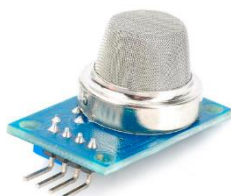


Figura 2 – Sensor de gás MQ-2.

Para a implementação do projeto na placa MSP430, utilizou-se os pinos analógicos A0 e A5, que na placa são descritos como P1.0 e P1.5, e os pinos digitais P1.1 e P1.3 da placa.

Os pinos analógicos foram utilizados a fim de fazer a leitura dos valores de tensão presentes na saída do sensor de gás e no circuito resistivo formado pelo LDR e um resistor (sendo que a resistência do LDR varia com a luz), sendo que esses valores são convertidos internamente no MSP430 (por um conversor A/D) e expressos em valores entre 0 e 1023. No sensor de gás, a tensão aferida pela porta varia a partir da concentração de gases

detectados, já no LDR a tensão varia com a luminosidade captada pelo mesmo.

Já os pinos digitais correspondem as entradas do LED e Buzzer, entradas as quais dependem dos valores de tensões, captados no LDR e sensor de gás, e aferidos pelas portas analógicas A0 e A5, respectivamente, de modo que as saídas digitais comecem em baixo (0) e vão para alto (1) quando um determinado valor de tensão for obtido.

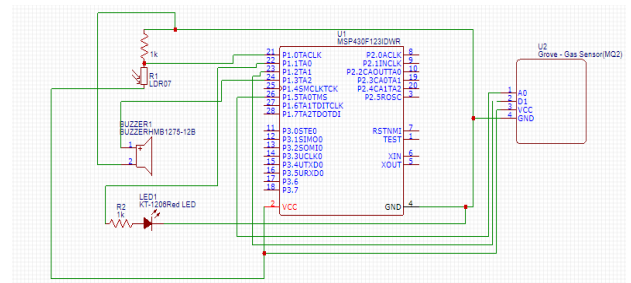


Figura 3 – Conexões com os sensores e o microcontrolador.

SOFTWARE

Inicialmente incluiu-se algumas bibliotecas e definiu-se algumas variáveis, além de definir algumas portas como entradas e saídas, bem como alguns valores iniciais de saída, como pode ser observado nas figuras abaixo.

```
1 #define LDR BIT0
2 #define LED BITS
3 #define BUZZER BIT2
4 #define LIMITE 100 //LIMITE PARA O NÍVEL DE GAS NOCIVO
5 volatile unsigned int i=0;
6 volatile unsigned GAS;
7 void inter();
8
```

Figura 4 – Definições de algumas variáveis.

```
9 int main(void)
10 {
11     WDTCTL = WDTPW | WDTHOLD; // DESLIGA watchdog timer
12     P1OUT &= ~(BUZZER + LED);
13     P1DIR |= (BUZZER + LED);
14     P1DIR &= ~(BIT1);
15     P1SEL |= (BUZZER + LED);
16
```

Figura 5 – Definições de entradas e saídas.

Como mencionado, o projeto utiliza um sensor analógico (sensor de gás nocivo) e uma resistência que varia com a luz (LDR), utilizados afim de identificar as situações exigidas. Para tais sinais serem reconhecidos pelo microcontrolador, foi necessário o uso de um conversor analógico-digital de 10 bits presente no MSP430 (ADC10), pois, é necessário ler os valores de tensão (sinal analógico) da saída do sensor de gás e no LDR, valores os quais variam para cada fator (gás nocivo e luz). Os valores de tensão podem ser lidos nas portas P1.0 à P1.5 do MSP430 utilizado.

Na figura 4, pode-se observar a conversão ADC realizada para a medição do sensor de gás, nos quais são selecionados

alguns de seus parâmetros, como o clock e a entrada analógica utilizada.

```
void Init_AD(void){
    ADC10CTL0 = ADC10SHT_2 + ADC10ON + ADC10IE; //ADC10IE - HABILITA INTERRUPTCAO
    ADC10CTL1 = INCH_1; // ENTRADA A1
    ADC10AE0 |= BIT1;
```

Figura 6 – Código para a conversão AD do sensor de gás.

Os valores obtidos através da conversão AD ficam armazenadas na memória, em um registrador específico do conversor (ADC10MEM). Assim, quando o microcontrolador identificar um nível maior que o limite pré-determinado de gás, o buzzer é acionado. Para isto, foi utilizada uma interrupção, como observada nas figuras 5 e 6.

```
26
27 _enable_interrupts();
28 for(;;){
29     ADC10CTL0 |= ENC + ADC10SC;
30     _bis_SR_register(CPUOFF + GIE);
31 }
```

Figura 7 – Definição de interrupção

```
34 #pragma vector=ADC10_VECTOR
35 __interrupt void ADC10_ISR (void)
36 {
37     ADC10CTL0 &= ~ADC10IFG;
38     GAS = ADC10MEM;
39     if (LIMITE <= GAS){
40         CCTL0 = CCIE;
41         CCR0 = 50000;
42         TACTL = TASSEL_2 + MC_1;
43         P1OUT |= BUZZER;
44         CCTL0 = ~CCIE;
45     }
46     else if (LIMITE > GAS){
47         P1OUT &= ~BUZZER;
48     }
49     __bic_SR_register_on_exit(CPUOFF);
50     delay_cycles(160000);
51     i++;
52 }
53 void inter()
54 {
55     P1IE = BIT1;
56     P1IES = BIT1;
57     P1IFG = 0x00;
58 }
59
```

Figura 8 – Interrupção realizada para o acionar o Buzzer.

REFERENCIAS BIBLIOGRÁFICAS

- [1] Arduomotive_com. Arduino smart working helmet. Disponível em: <http://www.instructables.com/id/Arduino-Smart-Working-Helmet/>
- [2] A. C. Ramos Gonçalves. Riscos associados a exploração mineira. Disponível em: http://www.uc.pt/fluc/depgeo/Cadernos_Geografia/Numeros_publicados/CadGeo30_31/Eixo1_9.
- [3] Oliveira OIT – Organização Nacional do Trabalho, Sobre saúde e segurança nas minas. Disponível em: <http://www.oitbrasil.org.br/content/sobre-seguran%C3%A7a-e-sa%C3%BAde-nas-minas.>
- [4] P. Cézanne Pinto. Avaliação das condições ambientais na mineração em subsolo. Disponível em: <http://www.scielo.br/pdf/rem/v59n3/v59n3a10>

- [5] FilipiFlop, Sensor de gás e fumaça MQ-2. Disponível em: <https://www.filipeflop.com/produto/sensor-de-gas-mq-2-inflamavel-e-fumaca/>
- [6] FritzenLab, Como funciona um LDR, Disponível em: <http://fritzenlab.com.br/2016/01/como-funciona-um-ldr-resistor-dependente-de-luz/>

ANEXOS

```
void setup() {
    analogReference(DEFAULT);
    //definição de entradas e pinos analogicos/digitais
    pinMode(3,OUTPUT);
    pinMode(5,OUTPUT);
    digitalWrite(3,LOW);
    digitalWrite(5,LOW);
}
```

```
void loop() {

    // carregando valores analogicos
    int ldrValue = analogRead(A0);
    int gasValue = analogRead(A5);
```

```
float voltage_ldr = ldrValue * (3.6 / 1023);
if (voltage_ldr < 0.03) {
    digitalWrite(3,HIGH);
}
else {
    digitalWrite(3,LOW);
}
if (gasValue > 90) {
    digitalWrite(5,HIGH);
}
else {
    digitalWrite(5,LOW);
}
}
```

