

Aluno: Rafael Teixeira de Oliveira

Disciplina: Software Embarcado

Data: 17/10/2018

Tarefa-04 – Apresentação das Tarefas 02,03 e 05



Sobre a aplicação

- ▣ Sensores usados:
 - Um joystick;
 - Um botão.
- ▣ Atuadores usados:
 - Um led;
 - Um servo motor.

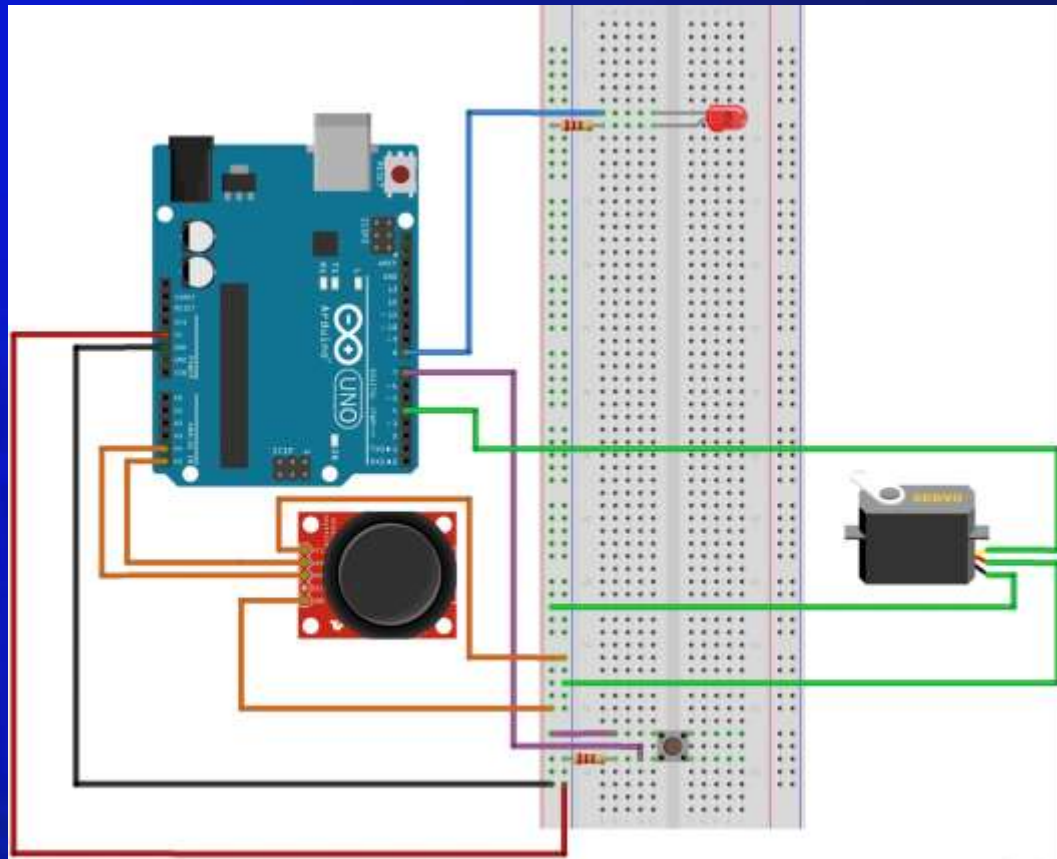


A Aplicação

□ O que faz a aplicação?

- Um botão, com função de “on/off”;
- Quando está no estado “on”:
 1. Recupera os valores do joystick (X e Y);
 2. Faz um cálculo com esses valores, cujo resultado é um inteiro, que pertence ao intervalo $[0, 180]$;
 3. Esse valor será a nova angulação do servo motor;
 4. Caso esse valor passe de um determinado limiar (90°), ativa o led;
 5. Caso contrário, deixa o led apagado.
- Ele fica em estado de “on” por um determinado espaço de tempo, no mínimo.

Organização física dos componentes





Código da Aplicação

```
#include <Servo.h>
```

```
Servo s;  
int v, estado_anterior = 1;  
float vx = 0, vy = 0;  
int soma;  
bool on_off = false;  
unsigned long c;
```

```
ISR(TIMER2_OVF_vect)    // rotina de interrupção  
{  
    if (on_off) {  
        c++;  
    }  
}
```



Código da Aplicação

```
void setup ()
{
  cli();
  DDRD &= ~(1 << 7); // Isto significa porta digital 7 servindo como
  entrada
  DDRB |= 1 << 0; // Isto significa porta digital 8 servindo como saída
  TCCR2A = 0b0;
  TCCR2B = 1 << CS21; // Modo Normal e Prescaler em 8
  TIMSK2 = 0b1;      // Habilita interrupção por overflow
  TCNT2 = 0;         // Reseta o TCNT2
  s.attach(4);
  s.write(0);
  sei();
  c = 0;
}
```



Código da Aplicação

```
void loop()
{
  testaOnOff();
  if (on_off) {
    vx = lerAnalogico(5);
    vy = lerAnalogico(4);
    soma = map(vx + vy, 0, 2048, 0, 180);
    s.write(soma);
    if (soma >= 90)
      PORTB |= 1 << 0; // Comando para fazer acender o led
    else
      PORTB &= ~(1 << 0); // Comando para deixar o led apagado
  }
}
```



Código da Aplicação

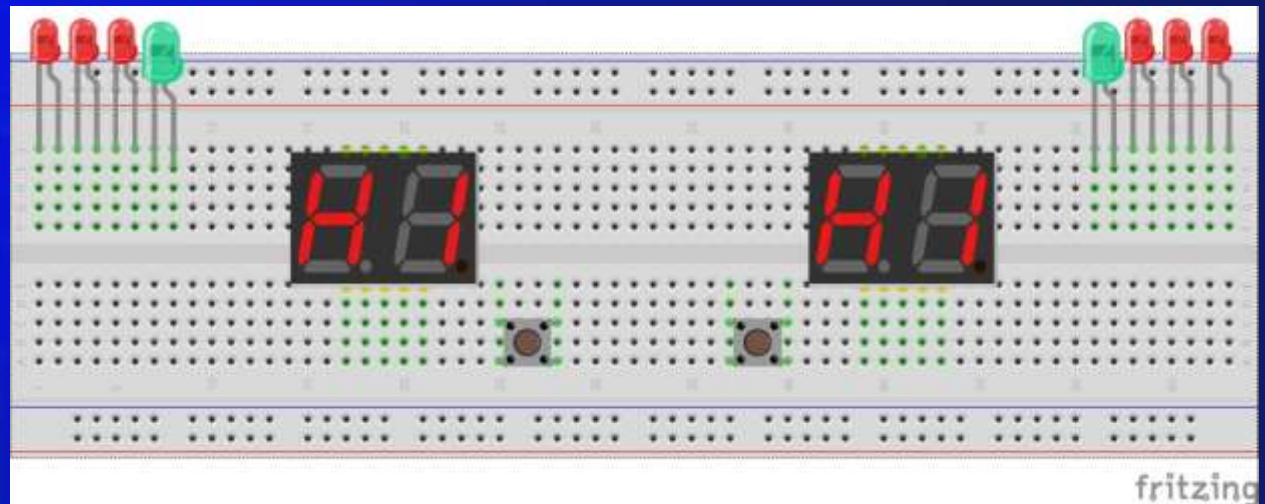
```
void testaOnOff() {  
    v = PIND >> 7; // Lendo o valor que está mapeado para o botão  
    if (!v) {  
        if (estado_anterior) {  
            if (on_off) {  
                if (c >= 125000) {  
                    on_off = !on_off;  
                    c = 0;  
                }  
            }  
        }  
        else {  
            on_off = !on_off;  
        }  
    }  
    estado_anterior = v;  
}
```




Código da Aplicação

```
int lerAnalogico(int porta){  
    ADMUX  = porta;           // para indicar porta analógica que será  
                               usada  
    ADMUX |= (1 << REFS0); // usa a voltagem pinada como referência  
    ADCSRA |= (1 << ADEN); // habilita o conversor analógico digital  
    ADCSRA |= (1 << ADSC);           // começa a conversão  
    while(ADCSRA & (1 << ADSC));    // espera terminar a conversão  
  
    int v = ADCL;               // pega os bits menos significativos  
    v = (ADCH << 8) + v;       // soma com os bits mais significativos  
    return v;  
}
```

Ideia para o projeto final





Obrigado!