

Aluno: Rafael Teixeira de Oliveira  
Disciplina: Software Embarcado  
Data: 19/12/2018  
Apresentação do Projeto Final



# O projeto proposto

- O projeto original: um placar eletrônico especializado em vôlei.
  - LEDs vermelhos: para marcar sets ganhos por cada equipe;
  - LEDs amarelos: para mostrar posse de bola;
  - Displays de 7 segmentos, de 2 dígitos cada: para mostrar o placar atual do set;
  - Quatro botões: para mandar comandos remotamente para aumentar ou diminuir o placar das equipes;
  - Para o funcionamento remoto seriam usados um transmissor e um receptor RF;
  - Para o funcionamento do projeto todo, seriam usados 2 Arduinos Uno.



# O projeto proposto

- Modificações no projeto em relação a proposta original:
  1. Troca dos 4 botões pelo controle remoto, fazendo com que não fosse mais necessário o uso de 2 Arduinos Uno;
  2. Troca do par transmissor e receptor RF pelo receptor IR.
- Nova versão do projeto usaria:
  - 8 LEDs (6 vermelhos e 2 amarelos);
  - 2 displays de 7 segmentos, de 2 dígitos cada;
  - 1 controle remoto;
  - 1 receptor de IR;
  - 1 circuito integrado especializado em displays;
  - 1 Arduino Uno.





# Versão final do projeto

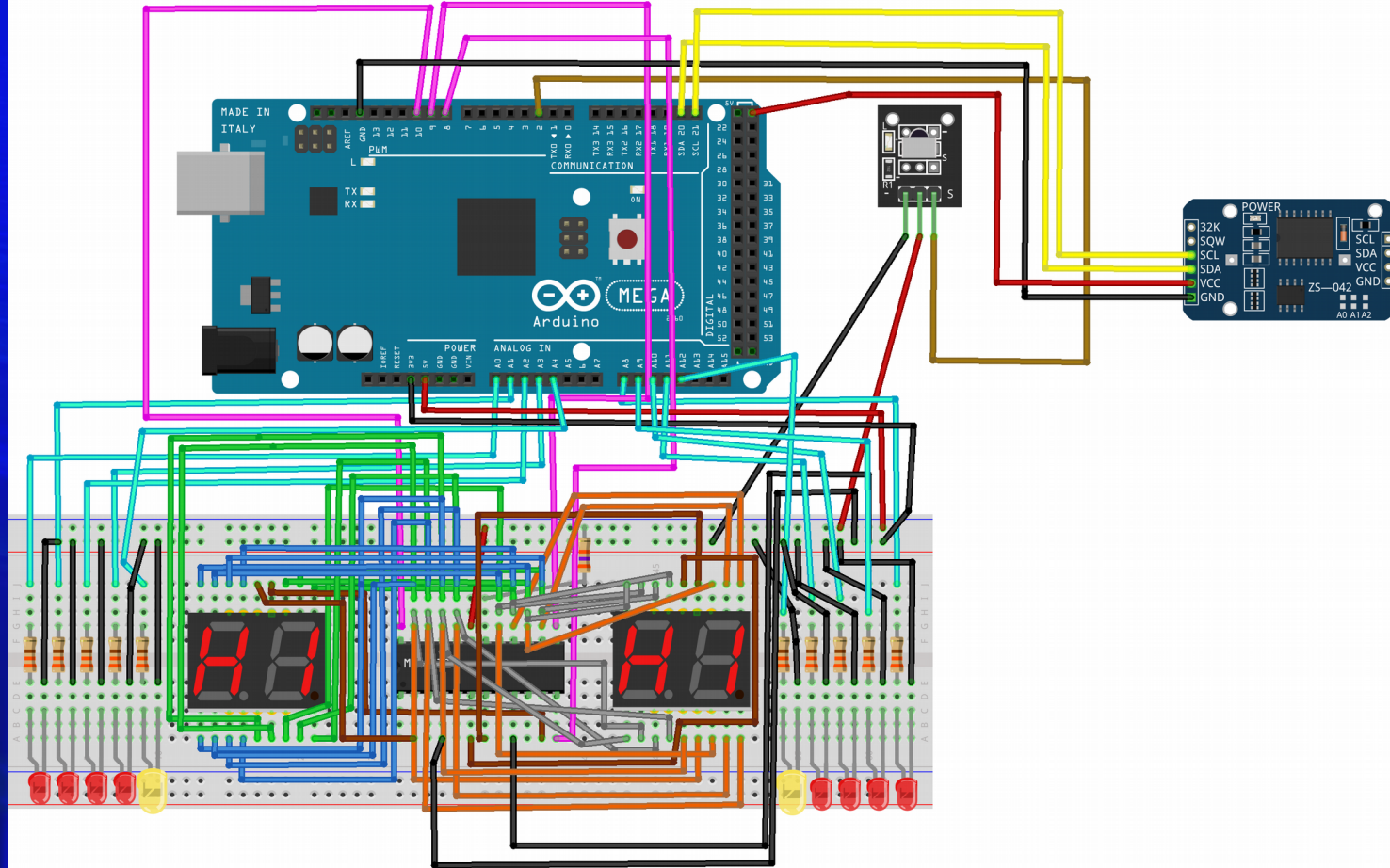
- A versão final do projeto, quando comparado com o escopo dito anteriormente, possui incrementos.
- Tais incrementos são tanto em termos computacionais, como em termos de hardware.
  - Em termos de hardware:
    - Foi adicionado um RTC para marcar tempo real de jogo;
    - Mais LEDs para ajudar a mostrar dados, como expulsos (futebol) e número de faltas em um período (basquete);
    - Troca do Arduino Uno pelo Arduino Mega.
  - Em termos computacionais:
    - Adições dos seguintes modos:
      - Futebol;
      - Basquete;
      - Cronômetro, e;
      - Contador Regressivo;



# Versão final do projeto

- Resumidamente:
  - O projeto final consiste em um placar eletrônico;
  - Tal placar pode registrar pontuação de jogos de vôlei, basquete e futebol;
  - Além disso, é possível marcar tempo decorrido e tempo restante através dos modos possíveis de cronômetro e contador regressivo;

# Organização física dos Componentes







# Código do projeto

```
#include <IRremote.h>
#include "LedControl.h"
#include "Volei.h"
#include "Futebol.h"
#include "Basquete.h"
#include "Cronometro.h"
#include "ContagemRegressiva.h"

volatile IRecv irrec(2);
volatile decode_results result;
LedControl lc = LedControl(10, 9, 8, 1);

int comando;
bool tem_comando;
Esporte* e = NULL;
```



# Código do projeto

```
void setup() {  
  // Equipe 1  
  pinMode(54, OUTPUT);  
  pinMode(55, OUTPUT);  
  pinMode(56, OUTPUT);  
  pinMode(57, OUTPUT);  
  pinMode(58, OUTPUT);  
  // Equipe 2  
  pinMode(62, OUTPUT);  
  pinMode(63, OUTPUT);  
  pinMode(64, OUTPUT);  
  pinMode(65, OUTPUT);  
  pinMode(66, OUTPUT);  
  irrec.enableIRIn();  
  tem_comando = false;  
  EICRB = (1 << ISC41) || (1 << ISC40);  
  EIMSK |= 1 << INT4;  
}
```





# Código do projeto

```
void loop() {  
  if (e == NULL) {  
    e = Inicializacao();  
  }  
  else {  
    if (e->situacao_jogo != 0) {  
      e->vitoria(lc);  
    } else {  
      e->Jogo(tem_comando, comando, lc);  
    }  
  }  
}
```



# Código do projeto

```
Esporte* Inicializacao() {  
    if (tem_comando) {  
        switch (comando) {  
            case 0xA25D: // CH-  
                return new Volei(lc);  
            case 0x629D: // CH  
                return new Futebol(lc);  
            case 0xE21D: // CH+  
                return new Basquete(lc);  
            case 0xC23D: // Play/Pause  
                return new Cronometro(lc);  
            case 0x906F: // EQ  
                return new ContagemRegressiva(lc);  
        }  
        tem_comando = false;  
    }  
    else {  
        return NULL;  
    }  
}
```



# Código do projeto

```
ISR(INT4_vect) {  
    sei();  
    if (!tem_comando) {  
        if (irrec.decode(&result)) {  
            tem_comando = true;  
            comando = result.value;  
            irrec.resume();  
        }  
    }  
}
```



# Código do projeto

```
void Volei::Jogo(bool& t, int& comando, LedControl lc) {  
    if (t) {  
        switch (comando) {  
            case 0x30CF:   
                pontoUm(1,lc);  
                break;  
            case 0x18E7:   
                pontoUm(-1,lc);  
                break;  
            case 0x7A85:   
                pontoDois(1,lc);  
                break;  
            case 0x10EF:   
                pontoDois(-1,lc);  
                break;  
        }  
        t = false;  
        comando = 0;  
    }  
}
```

```
void Futebol::Jogo(bool& t, int& comando, LedControl lc) {  
    if (t) {  
        switch (comando) {  
            case 0x30CF: // 1  
                if (tempo <= 2) gol(0,lc);  
                break;  
            case 0x18E7: // 2  
                if (tempo <= 2) tirarGol(0,lc);  
                break;  
            case 0x7A85: // 3  
                if (tempo <= 2) expulso(0);  
                break;  
            case 0x10EF: // 4  
                if (tempo <= 2) gol(1,lc);  
                break;  
            case 0x38C7: // 5  
                if (tempo <= 2) tirarGol(1,lc);  
                break;  
            case 0x5AA5: // 6  
                if (tempo <= 2) expulso(1);  
                break;  
            case 0xC23D: // Play/Pause  
                if (tempo <= 2) mostrarTempo(lc);  
                break;  
            case 0x906F: // Botão EQ  
                mudarTempo();  
                break;  
        }  
        t = false;  
        comando = 0;  
    }  
}
```

# Código do projeto

```
void Cronometro::Jogo(bool& t, int& comando, LedControl lc) {  
    if (t) {  
        switch (comando) {  
            case 0x30CF: // Parar cronômetro  
                funcionando = false;  
                break;  
            case 0x18E7: // Retomar cronômetro  
                if (!funcionando) {  
                    funcionando = true;  
                    clock.setDateTime(2018, 1, 1, horas, minutos, segundos);  
                }  
                break;  
            case 0x7A85: // Resetar cronômetro  
                clock.setDateTime(2018, 1, 1, 0, 0, 0);  
                mostrarTempo(lc);  
                break;  
        }  
        t = false;  
        comando = 0;  
    }  
    else {  
        if (funcionando)  
            mostrarTempo(lc);  
    }  
}
```