**FMU – FACULDADES METROPOLITANAS**

NATHAN RODRIGUES DE SOUZA - RA:6617073

**Atividade Prática Supervisionada - Arquitetura e Organização de Computadores**

São Paulo

2020

**NATHAN RODRIGUES DE SOUZA**

**Atividade Prática Supervisionada - Arquitetura e Organização de Computadores**

Trabalho Simulador de Semáforos em Arduino apresentado o professor Adilson Belluomini como requisito parcial para a obtenção do título Bacharel em Ciência da Computação.

São Paulo

2020

Sumário

[Introdução 4](#_Toc55638616)

[Desenvolvimento 5](#_Toc55638617)

[APS – Projeto Semáforos 6](#_Toc55638618)

[Referências Bibliográficas 8](#_Toc55638619)

[Apêndice – Código Arduíno 9](#_Toc55638620)

# Introdução

Uma plataforma eletrônica aberta baseada na construção de um ponte entre software e hardware de maneira simples, de maneira objetiva este é o Arduino. Permite desde a criação de simples projetos como o acender de luzes ,desenvolvimento de complexos sistemas de automação ou até mesmo *I.O.T*. Através de diferentes modelos, oferecem distintas funções bem como funcionalidades/capacidades.

# Desenvolvimento

Preocupado com a limitação que seus alunos de design de interação vivenciavam para o desenvolvimento de projetos derivado do alto preço das peças bem como parte de um conhecimento solido referente a eletrônica o professor Massimo Banzi junto a um grupo de 2 norte americanos, 2 italianos e um espanhol iniciam o projeto de uma plataforma para a simplificação entre a interação de software e hardware. Entretanto, com o passar do tempo esta iniciativa toma proporções cada vez mais elevadas, tornando assim possível a proximidade da área para qualquer entusiasta de maneira independente a seu nível de conhecimento quebrando assim paradigmas estruturados anteriormente (prévio além de solido conhecimento referente a eletrônica).

O projeto Arduino trata-se de uma plataforma open-source nas maior amplitude que o termo pode oferecer, as documentações, códigos bem com arquiteturas desenvolvidas pela equipe, futuramente empresa, são de livre acesso a todos possibilitando assim que estudantes do Chile, por exemplo, construam suas próprias placas gratuitamente, salvo claro as peças necessárias para sua fabricação. Operar de maneira livre permitiu a criação de inovadores projetos que contribuem não somente aos âmbitos acadêmicos mas igualmente em questões, por exemplo, janelas capazes de se regularem de acordo com a temperatura com o objetivo e facilitar os que possuem baixa mobilidade, unidades de alimentação para animais de estimação que detectam qual animal deverá ingerir qual alimento (caso algum sofra de enfermidades e necessite de dietas diferenciadas) além é claro de projetos como detectores de terremotos, aparelhos científicos de alto custo tornado assim equipamentos inacessível ferramentas uteis para o cotidiano.

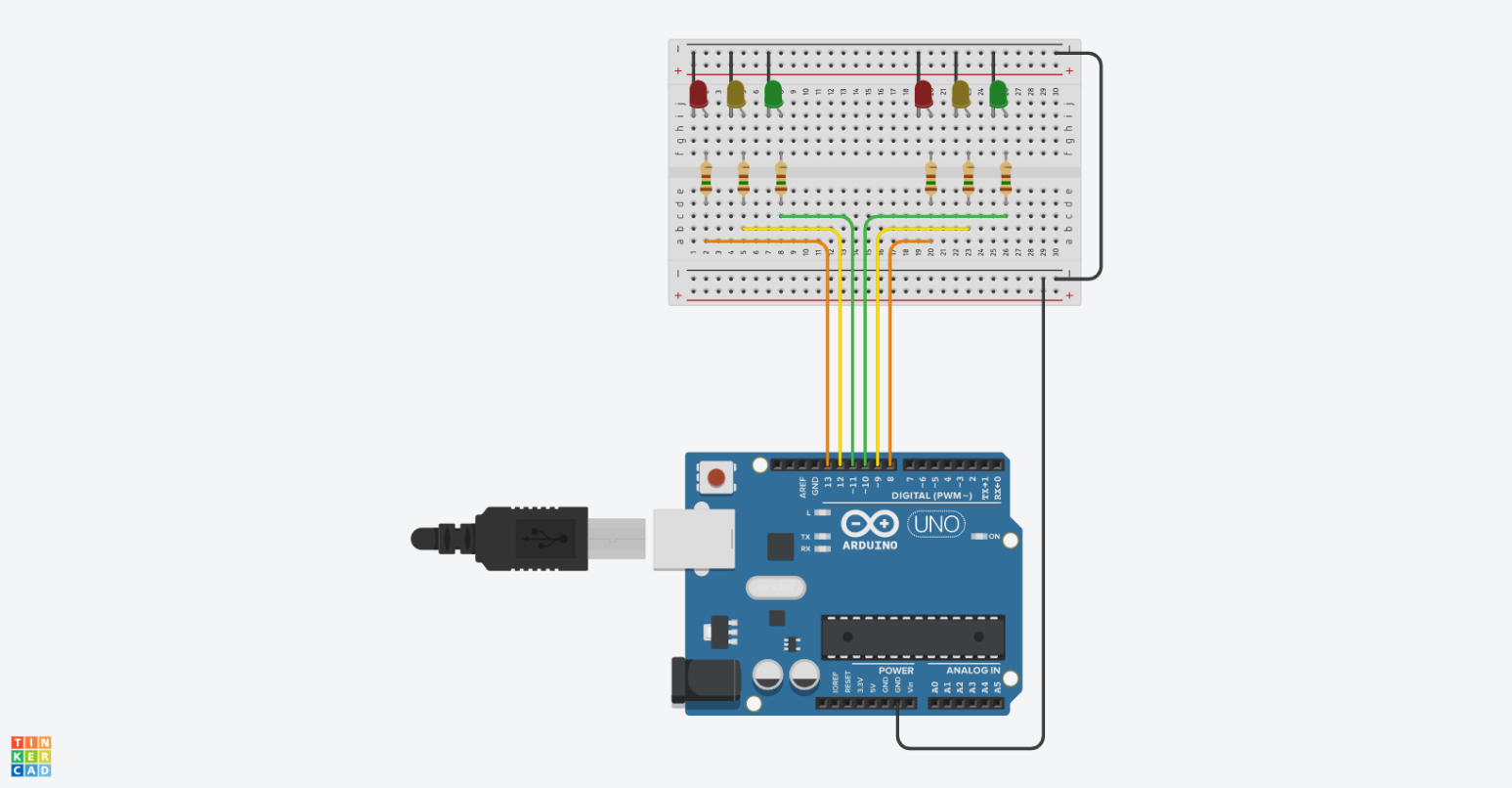
Tradicionalmente o modelo UNO do Arduino possui uma placa controladora bem como ambiente de desenvolvimento, sendo conhecimento como plataforma ao invés de um “mero” hardware. A placa por sua vez, possui uma serie de sensores, permitindo a interação com diversos dispositivos. A placa Arduino na versão Uno, é composta por uma entrada USB, entretanto pode ser alimentado por uma fonte externa, operando uma tensão que varia de 6 a 20 volts. Possui 14 pinos digitais de input e output sendo que 6 deles permitem PWM por padrão e 6 inputs analógicos, microcontrolador Atmega16U2. Abaixo maiores especificações técnicas referentes a placa em questão:

* Micro Controller ATmega328P
* Operating Voltage 5V
* Input Voltage (recommended) 7-12V
* Input Voltage (limit) 6-20V
* Digital I/O Pins 14 (of which 6 provide PWM output)
* PWM Digital I/O Pins 6
* Analog Input Pins 6
* DC Current per I/O Pin 20 mA
* DC Current for 3.3V Pin 50 mA
* Flash Memory 32 KB (ATmega328P) of which 0.5 KB used by bootloader
* SRAM 2 KB (ATmega328P)
* EEPROM 1 KB (ATmega328P)
* Clock Speed 16 MHz
* LED\_BUILTIN 13
* Length 68.6 mm
* Width 53.4 mm
* Weight 25 g

# APS – Projeto Semáforos

Em geral, o modelo Arduino Uno trata-se de um dos mais simples, entretanto por sua versatilidade permanece ótima opção, apresentando facilidade de conexão, alta extensão de projetos online, códigos abertos e exemplos de uso. Os modelos se diferenciam por suas características referentes a hardware, recursos disponíveis e o tamanho físico da placa, capacidade de processamento do microcontrolador, a memória, o clock speed, largura de banda, seu conjunto de recursos, e a finalidade de aplicação com relação ao tamanho do projeto físico, fator de suma importância.

Para a exemplificação das capacidades que esta ferramenta possui, um semáforo simulando a interação de duas ruas é desenvolvido. Ao início do código temos a criação de 6 variáveis g1, y1, r1 para o primeiro semáforo e g2, y2, r2 para o segundo, sendo elas usadas na função setup para a identificação de cada *LED* utilizado, funcionando de forma sincronizada através da utilização da função loop, responsável por executar todo o ciclo dos semáforos. Ao passo que as partes interajam entre si e operem de forma escalada, ou seja, após o semáforo 1 terminar seu ciclo o semáforo 2 inicie logo em seguida, utiliza-se o modo de funcionamento dos semáforos por tempo (não preemptivo), além do tempo do laço *FOR* nos *LEDs* amarelos dos dois semáforos que iram piscar quando estiverem próximos da ativação do próximo LED (Vermelho). Veja abaixo uma imagem ilustrativa do projeto:



Ver definição completa do projeto em:

https://www.tinkercad.com/things/0DXDBe06IzC-start%20simulating/editel?lessonid=EHD2303J3YPUS5Z&projectid=OIYJ88OJ3OPN3EA&collectionid=OIYJ88OJ3OPN3EA&sharecode=uFtQQSpCnX\_\_AKTI3DvoL8L0FrzE6DY0N6HdaCWI--I&sharecode=uFtQQSpCnX\_\_AKTI3DvoL8L0FrzE6DY0N6HdaCWI--I

# Referências Bibliográficas

|  |
| --- |
| https://arduinoaprendizes.wordpress.com/2015/04/22/historiaarduino/ |
| https://fabiocosta.net/arduino/historia-do-arduino/ |
| https://store.arduino.cc/usa/arduino-uno-rev3 |
| https://www.arduino.cc/en/Guide/Introduction |
| https://www.opservices.com.br/o-que-e-o-arduino/ |
| https://www.usinainfo.com.br/arduino-74 |
| https://www.youtube.com/watch?v=D4D1WhA\_mi8 |
| https://www.youtube.com/watch?v=eFCk3qWmCoo |
| https://www.youtube.com/watch?v=HvkDW9s1AnI&t=599s |

# Apêndice – Código Arduíno

int g1 = 11; //Led verde semáforo 1

int y1 = 12; //Led amarelo semáforo 1

int r1 = 13; // Led vermelho semáforo 1

int g2 = 10; //Led verde semáforo 2

int y2 = 9; //Led amarelo semáforo 2

int r2 = 8; // Led vermelho semáforo 2

void setup(){

pinMode(g1 , OUTPUT);

pinMode(y1 , OUTPUT);

pinMode(r1 , OUTPUT);

pinMode(g2 , OUTPUT);

pinMode(y2 , OUTPUT);

pinMode(r2 , OUTPUT);

}

void loop(){

digitalWrite(g1, HIGH);

digitalWrite(r1, LOW);

digitalWrite(g2, LOW);

digitalWrite(r2, HIGH);

digitalWrite(y1 , LOW);

delay(4000);

digitalWrite(g1, LOW);

digitalWrite(y1, HIGH);

delay(2000);

for(int x = 0 ; x <5; x++){// Led amarelo piscando

digitalWrite(y1, HIGH);

delay(400);

digitalWrite(y1, LOW);

delay(400);

}

digitalWrite(g2, HIGH);

digitalWrite(r2, LOW);

digitalWrite(g1, LOW);

digitalWrite(r1, HIGH);

digitalWrite(y1 , LOW);

delay(4000);

digitalWrite(g2, LOW);

digitalWrite(y2, HIGH);

delay(2000);

for(int x = 0 ; x <5; x++){ // Led amarelo piscando

digitalWrite(y2, HIGH);

delay(400);

digitalWrite(y2, LOW);

delay(400);

}

}