

Uso de Arduino para Irrigação Automatizada de Baixo Custo



Irrigação Automatizada de Hortas Urbanas

Aplicações

- Agricultura urbana
- Hortas residenciais
- Sistemas Agroflorestais

“Uso de Arduino para Irrigação Automatizada de Baixo Custo”, de Saulo Machado Jacques pode ser usado, compartilhado e modificado livremente, desde que citada a fonte, segundo a Licença **Creative Commons** - Atribuição 4.0 Internacional



Irrigação Automatizada de Hortas Urbanas

Potencial

- Integração com projetos de captação de água da chuva
- Uso eficiente de água
- Irrigação de pequenas e médias produções
- Adequação a necessidades e condições climáticas específicas de diferentes regiões
- Adequação a demanda hídricas de diferentes grupos vegetais

Irrigação Automatizada de Hortas Urbanas

A adoção dessa técnica não visa a substituição do contato das pessoas com a terra por um sistema 100% automatizado, mas um equilíbrio entre o resgate da agricultura em grandes centros urbanos, integrado ao uso eficiente de recursos naturais.

Irrigação Automatizada de Hortas Urbanas

Componentes

Arduino

Protoboard

Cabo conexão com laptop

Relé (relay)

Válvula solenóide

Mangueira

Ferramentas

Fios

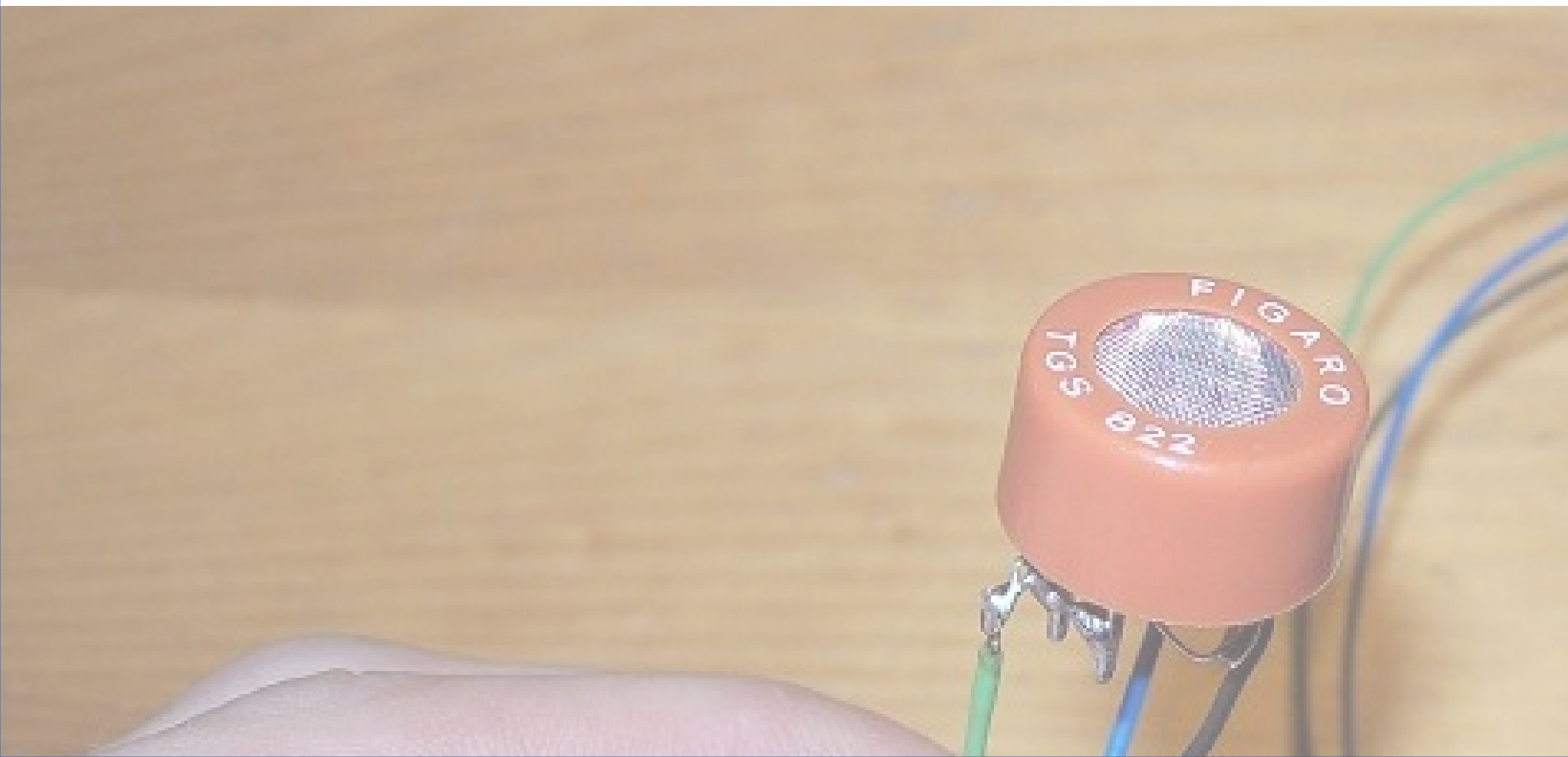
Fita isolante

Chave de fenda

Estilete

Algodão (ou terra)

Componentes



Irrigação Automatizada de Hortas Urbanas

O Relé (ou Relay)



**Circuito elétrico usado para fechar/abrir
circuito sem necessidade de intervenção**

Irrigação Automatizada de Hortas Urbanas

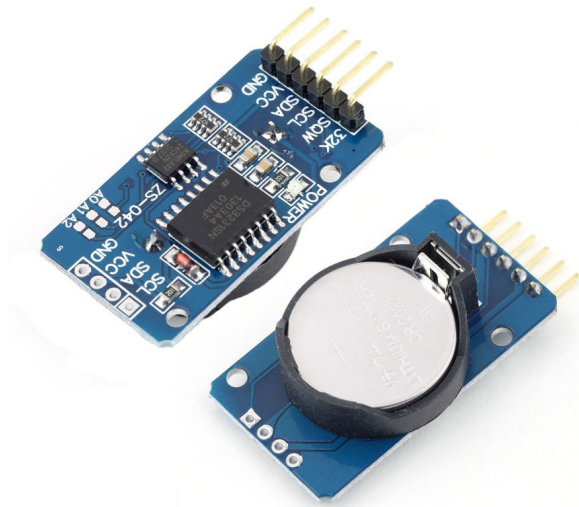
Válvula Solenoide



Válvula que se abre e se fecha respondendo ao campo magnético gerado por um solenoide ou um eletroímã

Irrigação Automatizada de Hortas Urbanas

Módulo RTC



Relógio de tempo real

Bateria que garante autonomia para fornecer informações de segundo, minutos, dia, data, mês e ano.

Medindo a Umidade Usando Arduino

O sensor utilizado funciona como um resistor que varia em relação à quantidade de água no solo: quanto mais água, maior condutividades e menor resistência.

Resultados obtidos com os sensores: Resistência

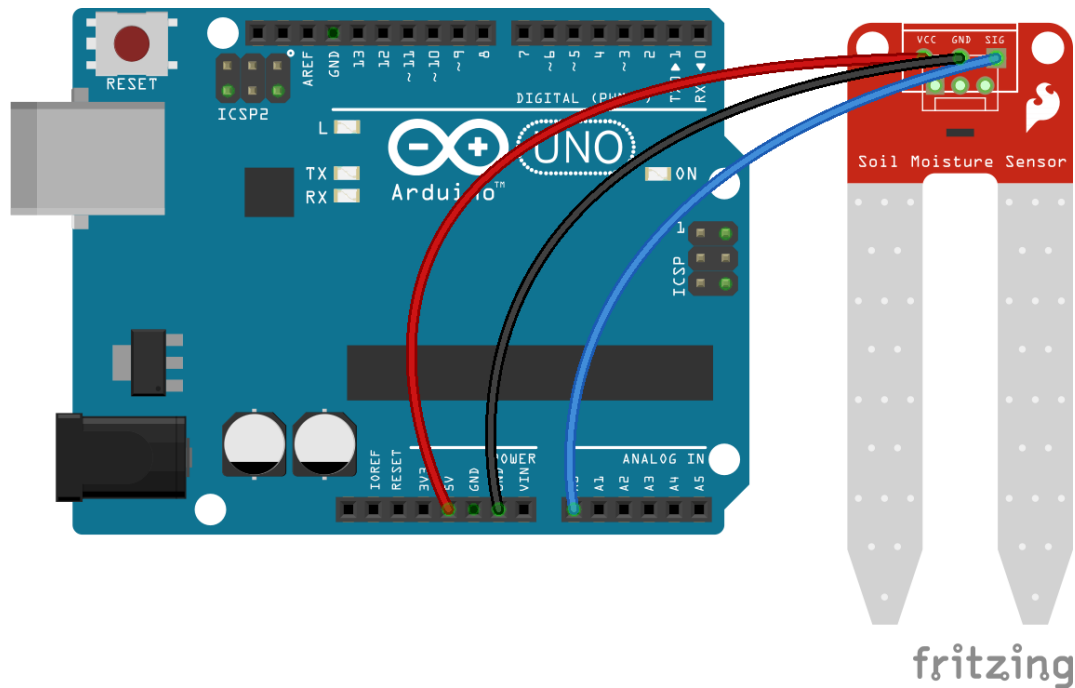
Alta concentração de água → Menores valores de output

Baixa concentração de água → Maiores valores de output

Lendo a Umidade do Solo com Arduino

Arquivo: esquema1_automatizada.ino

Irrigação Automatizada de Hortas Urbanas



```
int val_umid = 0; // sensor de umidade
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void loop() {  
  int umidade = analogRead(val_umid);
```

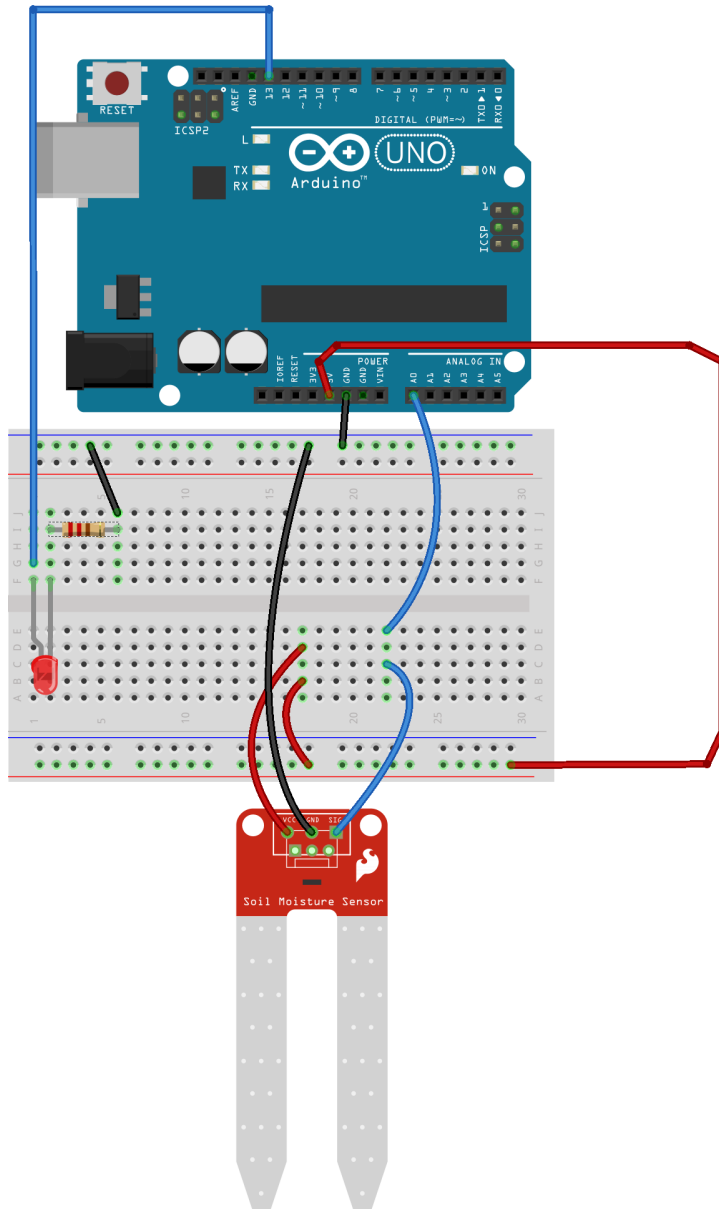
```
//convertendo o valor bruto em % de umidade  
umidade = constrain(umidade,400,1023);  
umidade = map(umidade,400,1023,100,0);
```

```
Serial.println("Umididade (valor bruto)");  
Serial.println(val_umid);  
Serial.println("Umididade (%)");  
Serial.println(umidade);  
}
```

Detectando Baixa Umidade do Solo

Arquivo: esquema2_led.ino

Irrigação Automatizada de Hortas Urbanas

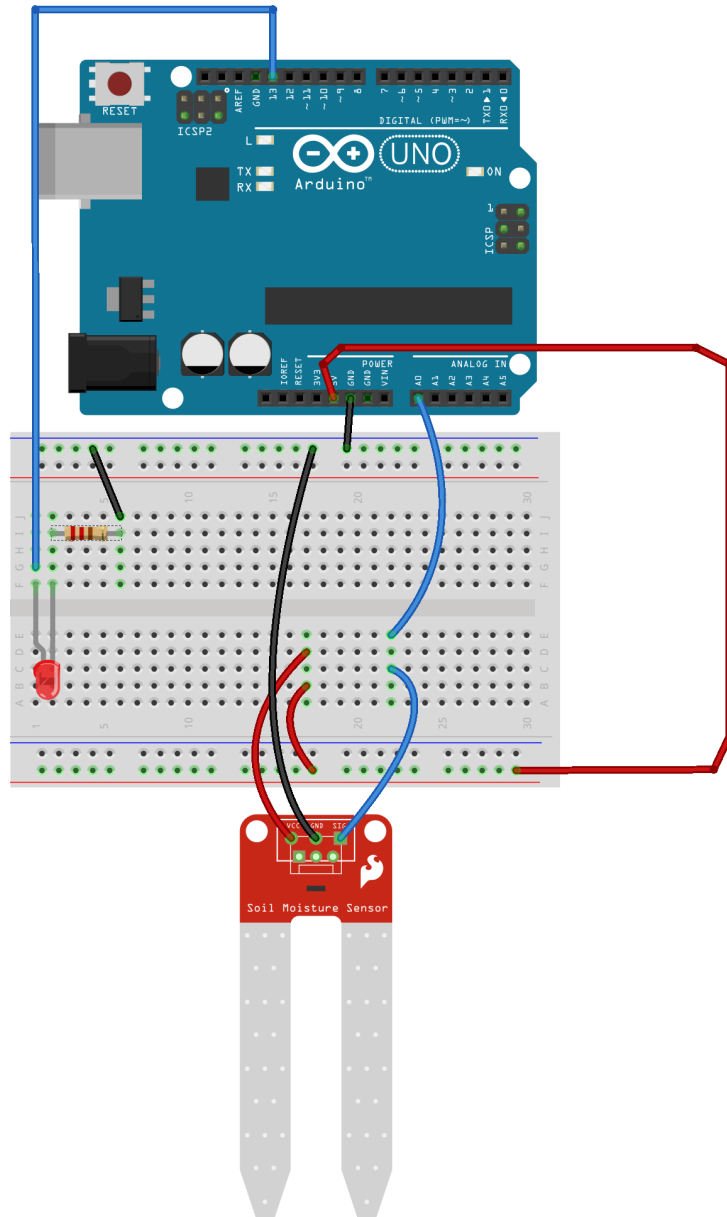


```
int val_umid = 0; // sensor de umidade  
int umidade=0;
```

```
void setup() {  
  Serial.begin(9600);  
}
```

```
void led(int state) {  
  digitalWrite(8, state);  
}
```

Irrigação Automatizada de Hortas Urbanas

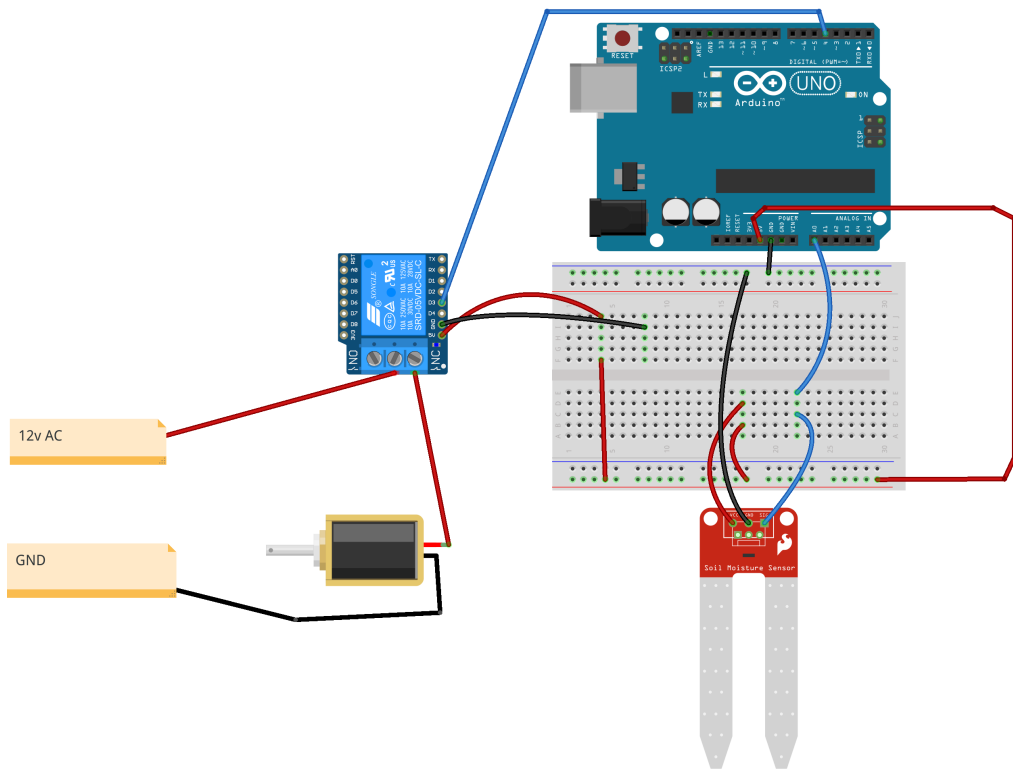


```
void loop() {  
  umidade = analogRead(val_umid);  
  
  //intervalo dos resultados brutos entre seca e  
  úmido  
  umidade = constrain(umidade,400,1023);  
  
  //convertendo os resultados em porcentagem  
  de umidade (0 e 100%)  
  umidade = map(umidade,400,1023,100,0);  
  
  Serial.println(umidade);  
  
  if (umidade< 40) {  
    led(HIGH); }  
  else {  
    led(LOW); }  
  
  delay(2000);  
}
```


Detectando Baixa Umidade do Solo & Automatizando o Sistema

Arquivo: esquema3_automatizada.ino

Irrigação Automatizada de Hortas Urbanas

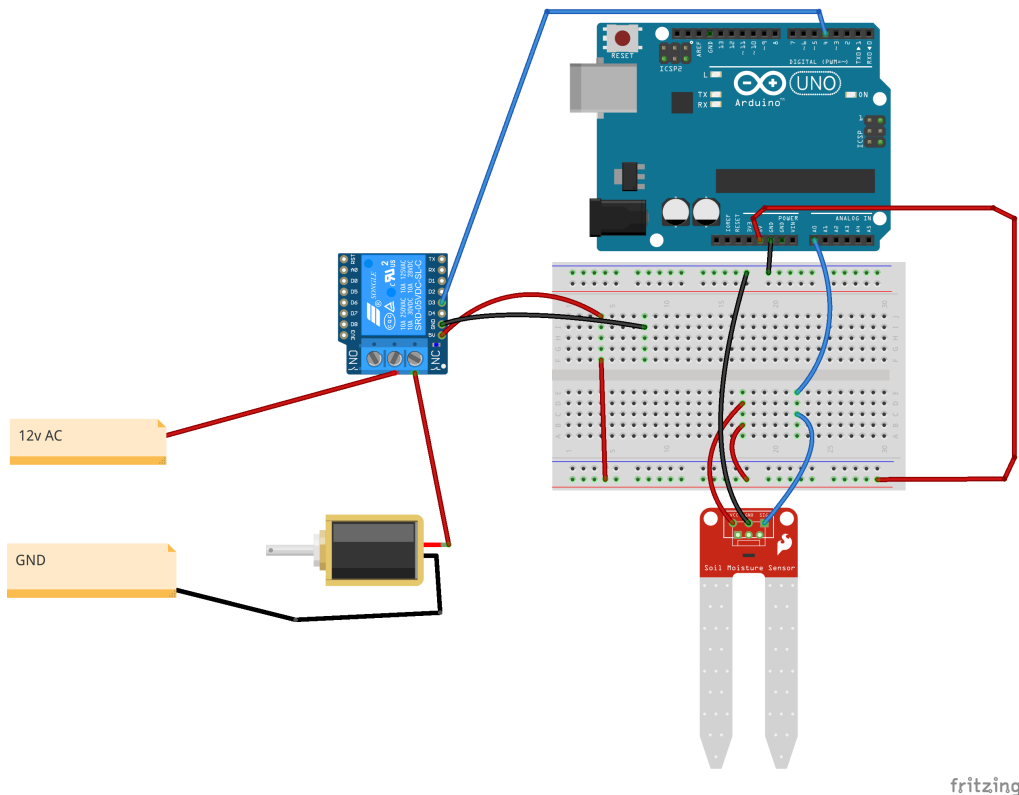


```
int val_umid = 0; // sensor de umidade
int solenoide = 8; // porta ativa a solenoide
```

```
void setup() {
  Serial.begin(9600);
}
```

```
void rega(int state) {
  digitalWrite(solenoide, state);
}
```

Irrigação Automatizada de Hortas Urbanas



```
void loop() {  
  int umidade = analogRead(val_umid);  
  umidade = constrain(umidade,400,1023);  
  umidade = map(umidade,400,1023,100,0);
```

```
  Serial.println(umidade);
```

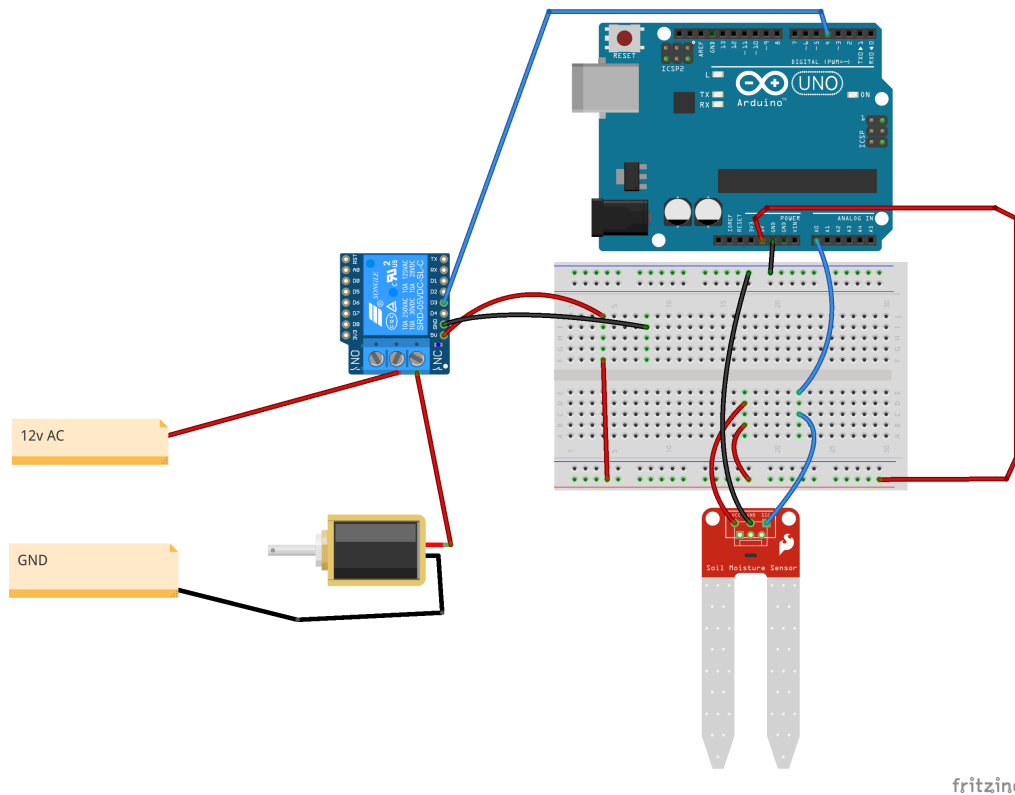
```
  if(umidade< 40) {  
    rega(HIGH);  
  } else {  
    rega(LOW);  
  }  
  delay(2000);
```

```
}
```

Usando Tempo Para Controlar a Rega

Arquivo: esquema4_tempo_de_rega.ino

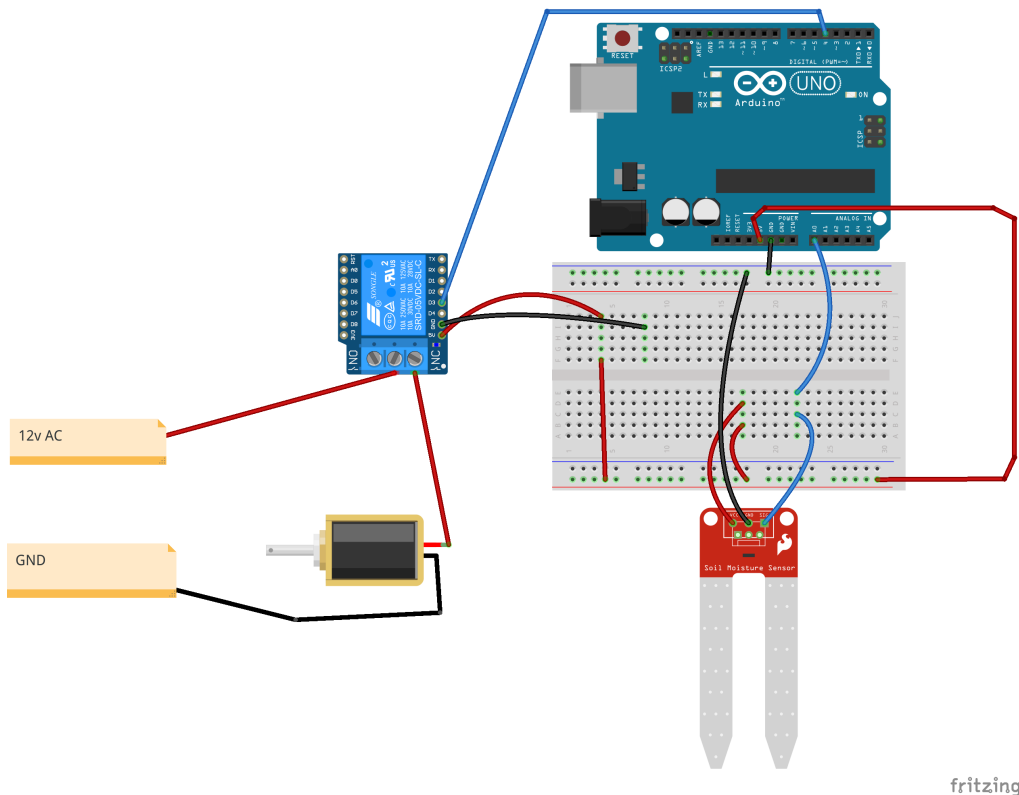
Irrigação Automatizada de Hortas Urbanas



```
int val_umid = 0; // sensor de umidade
int umidade=0;
int solenoide = 8; // ativar led quando seco
int TempoRega = 10000; //definir tempo de rega (10 secs operando)
```

```
void setup() {
  Serial.begin(9600);
  pinMode(solenoide, OUTPUT); //pin do solenoide
  digitalWrite(solenoide, LOW); //certifica-se que a solenoide inicia deligada (LOW)
}
```

Irrigação Automatizada de Hortas Urbanas



```
void loop() {  
  umidade = analogRead(val_umid);  
  umidade = constrain(umidade,400,1023);  
  //intervalo entre seca e úmido  
  umidade = map(umidade,400,1023,100,0);
```

```
  Serial.println(umidade);
```

```
  if (umidade < 40) {  
    digitalWrite(solenoid, HIGH);  
    delay(TempoRega);  
    digitalWrite(solenoid, LOW);  
  }  
}
```

Incluindo os Dados do RTC

Arquivo: esquema5_RTC.ino

Criando Nosso Protótipo (v0.1_esdi)

Arquivo: esquema5_RTC.ino

Irrigação Automatizada de Hortas Urbanas

```
// Carrega a biblioteca virtuabotixRTC  
#include <virtuabotixRTC.h>
```

```
//Sensor de umidade  
int val_umid = 0;  
int umidade=0;
```

```
// Valvula solenoide  
int solenoide = 8; // ativar led quando seco
```

```
// Determina os pinos ligados ao modulo  
// myRTC(clock, data, rst)  
virtuabotixRTC myRTC(10, 9, 8);
```

```
const int H1 = 17; // hora medida 1  
const int M1 = 07; // minuto medida 1
```

```
const int H2 = 17; // hora medida 2  
const int M2 = 11; // minuto medida 2
```

```
// Definindo umidade limite e tempo de rega  
const int umidade_min = 40; // 80%
```

```
int TempoRega = 10000; //10 secs operando
```

Irrigação Automatizada de Hortas Urbanas

```
void setup() {  
    Serial.begin(9600);  
  
    //certificar que a solenoide inicia deligada (LOW)  
    digitalWrite(solenoide, LOW);  
  
    //definir solenoide como saida  
    pinMode(solenoide, OUTPUT);  
}
```

Irrigação Automatizada de Hortas Urbanas

```
void loop()
{
  // Ler as informacoes do CI
  myRTC.updateTime();
  imprime_tempo();

  int hora = myRTC.hours;
  int minutos = myRTC.minutes;

  // verifica se chegou a hora de ler a umidade
  if ((hora == H1 && minutos == M1) || (hora == H2 && minutos == M2) )
  {
    Serial.println("medindo...");
  }
}
```

Irrigação Automatizada de Hortas Urbanas

```
// ler umidade
  umidade = analogRead(val_umid);
//intervalo entre seca e úmido
  umidade = constrain(umidade,400,1023);
// convertendo em %
  umidade = map(umidade,400,1023,100,0);
  imprime_umidade();

// verificar se a umidade está abaixo do minimo
  if (umidade < umidade_min ) {
    Serial.println("abrindo válvula...");
    digitalWrite(solenoide, HIGH);
    delay(TempoRega);
    Serial.println("fechando valvula...");
    digitalWrite(solenoide, LOW);
  } // fim do if da umidade
}
}
```

Irrigação Automatizada de Hortas Urbanas

```
void imprime_umidade() {  
    Serial.print("Umidade : ");  
    Serial.print(umidade);  
    Serial.println("%");  
}
```

Irrigação Automatizada de Hortas Urbanas

```
void imprime_tempo() {  
    // Imprime as informacoes no serial monitor  
    Serial.print("Data : ");  
    Serial.print(myRTC.dayofmonth);  
    Serial.print("/");  
    Serial.print(myRTC.month);  
    Serial.print("/");  
    Serial.print(myRTC.year);  
    Serial.print(" ");  
    Serial.print("Hora : ");
```


Irrigação Automatizada de Hortas Urbanas

```
// Adiciona um 0 caso o valor da hora seja <10
if (myRTC.hours < 10)
{
    Serial.print("0");
}
Serial.print(myRTC.hours);
Serial.print(":");
// Adiciona um 0 caso o valor dos minutos seja <10
if (myRTC.minutes < 10)
{
    Serial.print("0");
}
Serial.print(myRTC.minutes);
Serial.print(":");
// Adiciona um 0 caso o valor dos segundos seja <10
if (myRTC.seconds < 10)
{
    Serial.print("0");
}
Serial.println(myRTC.seconds);
}
```