

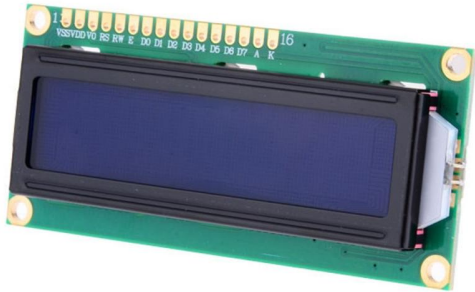
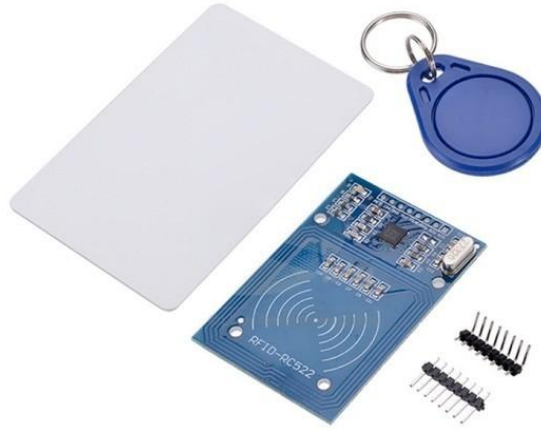
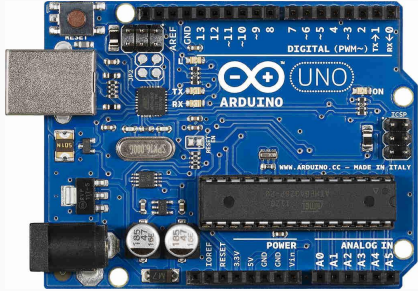
IT'S TIME TO D-D-D-DUEL

Gustavo Silva
Luiz Andrade

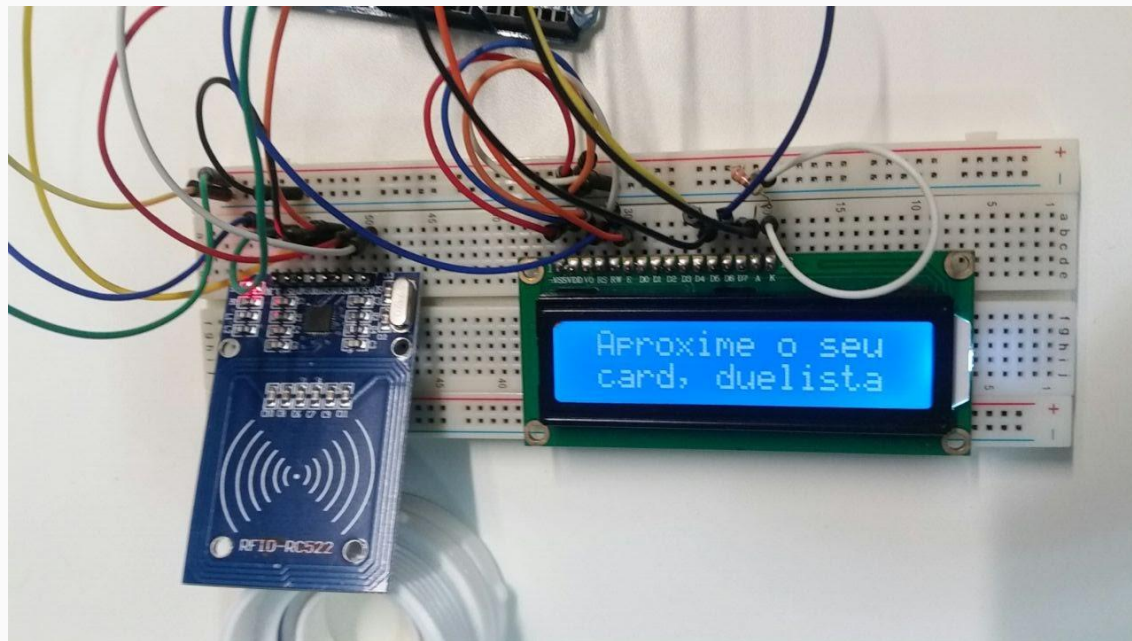
Objetivo

- Aproveitar projeto de validação RFID
- Integrar Arduino-Unity
 - Ler informações dos cartões
 - Utilizar a serial port para receber/enviar info

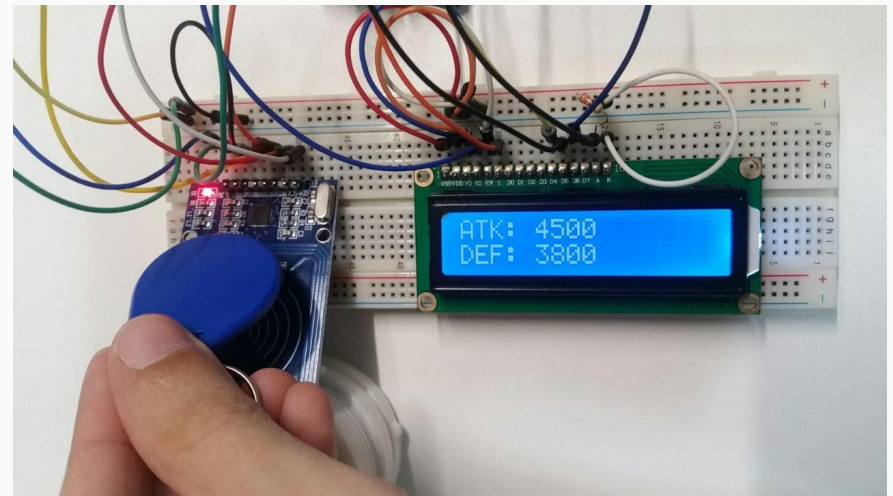
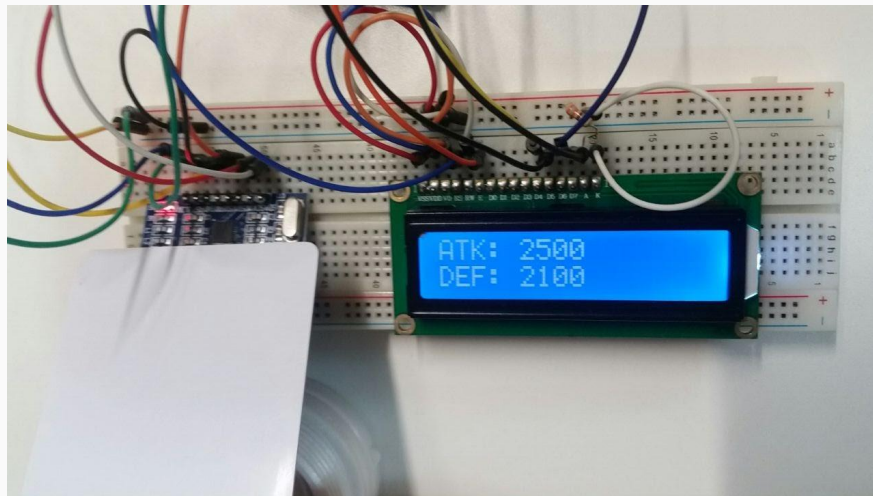
Componentes



Funcionamento



Funcionamento



Funcionamento



Código/Arduino

```
void setup() {(...)} //Detalhes do Setup
```

```
void loop()
```

```
{  
    if (Serial.available() > 0)  
    {  
        sCmd.readSerial();  
    }  
}
```

```
if(!fixed && millis()-count >= 3000)  
{  
    msgInicial();  
    fixed = true;  
}
```

```
if(!mfrc522.PICC_IsNewCardPresent()) // Procura por novos cartões  
    return;
```

```
if(!mfrc522.PICC_ReadCardSerial()) // Aceita somente se um cartão for lido por vez  
    return;
```

```
String conteudo = "";
```

```
char conteudoChar[50];
```

```
for(byte i = 0; i < mfrc522.uid.size; i++)  
    conteudo.concat(String(mfrc522.uid.uidByte[i]));
```

```
if(validar(conteudo))  
{  
    conteudo += '\n';  
    conteudo.toCharArray(conteudoChar, 50);  
    Serial.write(conteudoChar);  
}
```

```
count = millis();
```

```
}
```

Código/Arduino

```
bool validar(String conteudo)
{
    lcd.clear();
    lcd.setCursor(0,0);

    if(conteudo == "2297012136") //UID 1 - Chaveiro - Dragao Branco
    {
        lcd.print("ATK: 4500");
        lcd.setCursor(0,1);
        lcd.print("DEF: 3800");
        fixed = true;
        return true;
    }
    else if (conteudo == "1099148229") //UID 2 - Cartao - Mago Negro
    {
        lcd.print("ATK: 2500");
        lcd.setCursor(0,1);
        lcd.print("DEF: 2100");
        fixed = true;
        return true;
    }
}
```

```
//Cartões desconhecidos
lcd.print("  Carta Falsa");
lcd.setCursor(0,1);
lcd.print("  OBLITERADO!");
fixed = false;
return false;
}
```


Código/Unity

```
//Main.cs
```

```
private void StartConnection()  
{  
    con = new ConnectionHandler(port);  
    con.Start();  
}
```

```
//ConnectionHandler.cs
```

```
public void Start()  
{  
    outputQueue = Queue.Synchronized( new Queue() );  
    inputQueue  = Queue.Synchronized( new Queue() );  
    thread = new Thread(ThreadLoop);  
    thread.Start();  
}
```

Código/Unity

```
public void ThreadLoop(){
    stream = new SerialPort(port, baudrate);
    stream.ReadTimeout = 50;
    stream.Open();
    string stringResult = "";
    while (IsLooping()){
        // Send to Arduino
        if (outputQueue.Count != 0){
            string command = (string) outputQueue.Dequeue();
            WriteToArduino(command);
        }
        // Read from Arduino
        char charResult = ReadFromArduino(stream.ReadTimeout);
        if (charResult != 'e'){
            if(charResult == '\n'){
                inputQueue.Enqueue(stringResult);
                stringResult = "";
            }else{ stringResult += charResult; }
        }
    }
    stream.Close();
}
```

```
private void WriteToArduino(string message){
    stream.WriteLine(message);
    stream.BaseStream.Flush();
}

private char ReadFromArduino (int timeout = 0)
{
    stream.ReadTimeout = timeout;
    try
    {
        return (char) stream.ReadByte();
    }
    catch(TimeoutException){
        return 'e';
    }
}
```

Demonstração do programa

