ABNT

# Tutorial de comunicação via internet para o manipulador robótico

**Autor:**
Gabriel Araujo Zucchi

## 1 Introdução

Este documento têm como objetivo orientar o leitor de como realizar a comunicação entre controlador e um computador com o objetivo de integrar o manipulador robótico a internet. O tutorial será divido em duas parte Setup e Programação.Em setup será instruído como realizar a conexão e configurações previas para utilização dos módulos de WebSocket.Estes módulos serão descritos na parte de Programação de um modo genérico para instruir como começar uma conexão simples com um cliente ou um servidor.

## 2 Setup

1. Simulação em Robot Studio

A única configuração necessária para a simulação é a habilitação de "PC-Interface".Para isso é necessário mudar alterar esta configuração no controlador simulado nas opções do controlador.Para acessar a janela de opções do controlador, clique com o botão direito do mouse no nome do controlador criado quando aberto o robot studio e selecione "Change Option"conforme a figura a seguir mostra:

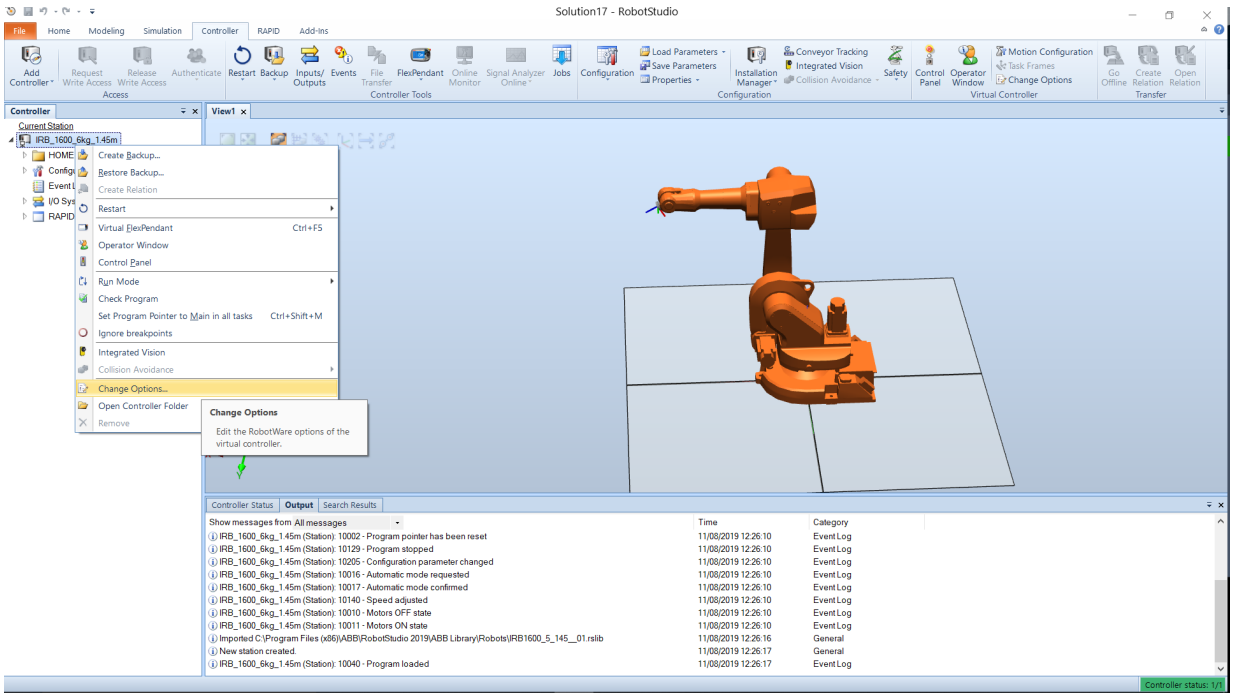


*Figura 1: Acesso as opções do controlador*

Ao clicar nessa opção a janela de opções se abrirá,selecione "Communication"e habilite a opção "PC Interface":

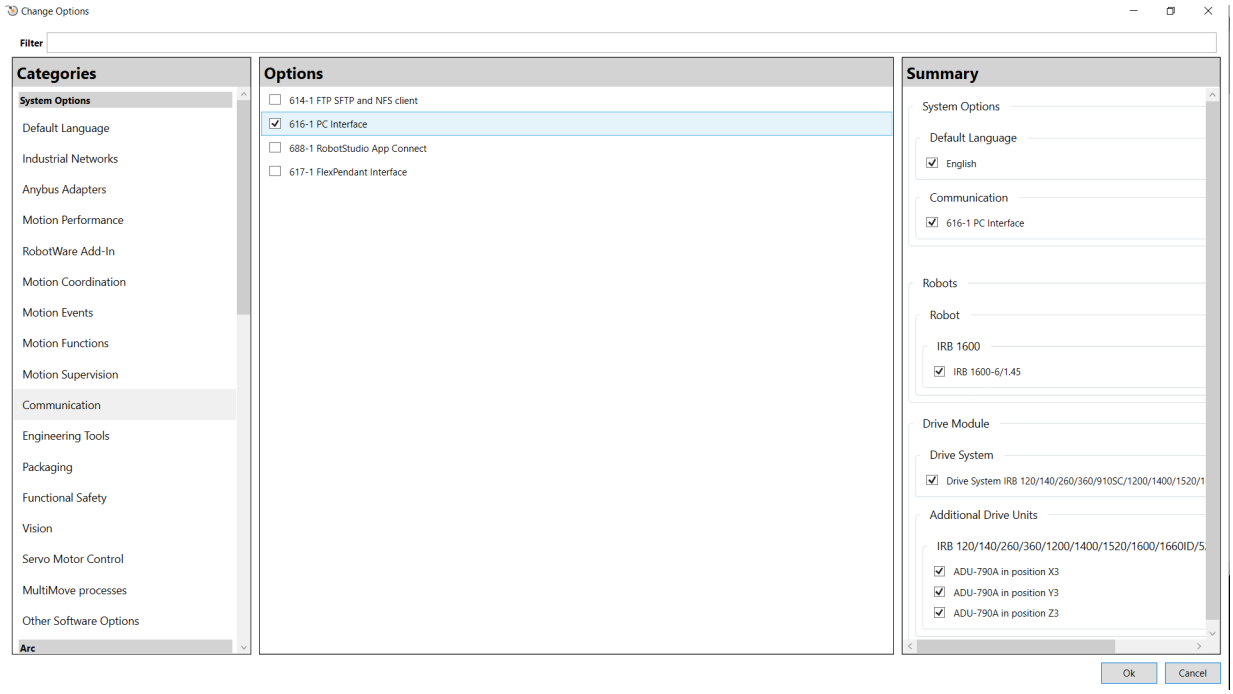


*Figura 2: Habilitação de PC Interface*

Com isso todas as funções de WebSocket podem ser utilizadas para a simulação.

2. Controlador Físico

Primeiramente é necessário conectar o cabo ethernet na porta disponível no controlador e na outra ponta em algum computador com RobotStudio instalado.Com isso será possivel abrir o controlador fisico através do RobobtStudio através de FILE>ONLINE>ONE-CLICK-CONNECT.Assim seu computador será conectado a rede LAN(Local Access Network) fornecido pelo controlador.

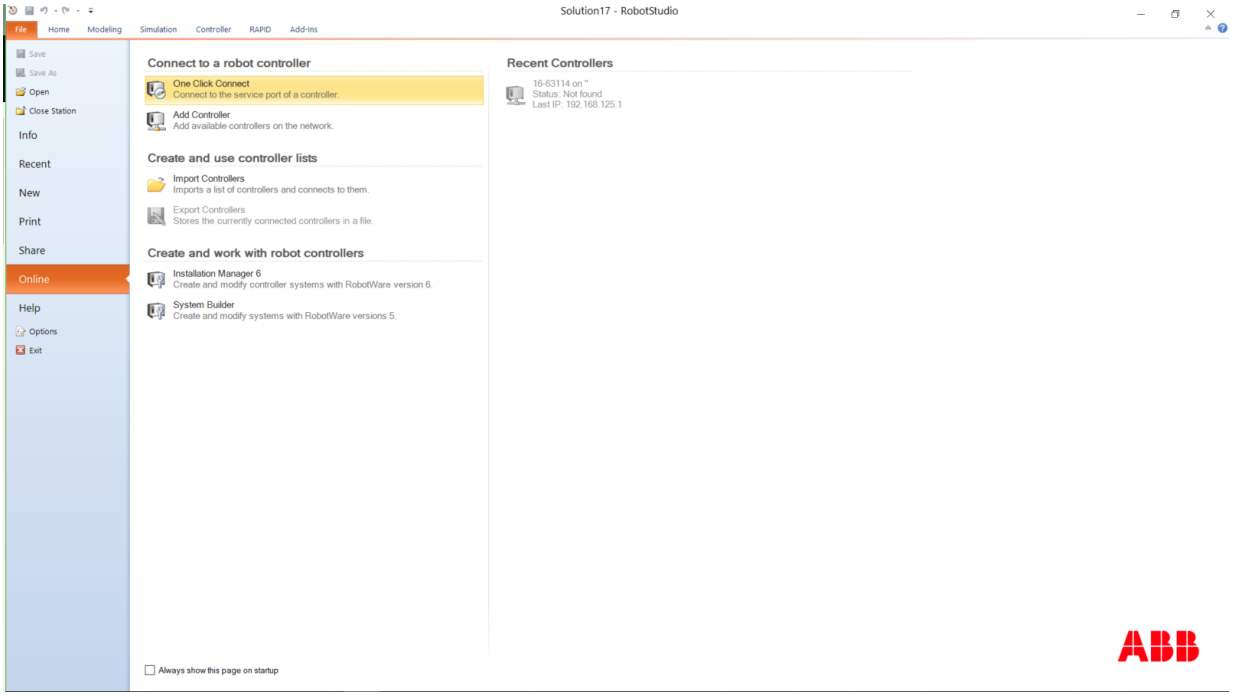


*Figura 3: One Click Connection*

Se tudo ocorreu como o esperado, uma janela abrirá onde será possivel escolher o controlador em qual o computador está conectado e abri-lo no RobotStudio.Por fim rode seu programa atrvés do FlexPendant normalmente que todas as funções de WebSocket devem funcionar corretamente.

Outras formas de conectar o computador com o controlador podem ser encontrada em [robotstudio 2019]

# 3 Programação

Nesta seção será descrita como iniciar um cliente e um servidor escrito em Python através de WebSocket para utilização no RobotStudio.

Vamos iniciar pela criação de um cliente na linguagem RAPID.Esse cliente irá se comunicar com um server exterior, podendo este ser em Node.js, Python, outro controlador ou qualquer linguagem que seja

```
1   MODULE Module1
2       PROC main()
3           VAR socketdev client;
4           VAR rawbytes data;
5           SocketCreate client;
6           SocketConnect client,"127.0.0.1",8080;
7           SocketSend client, \Str := "Teste";
8           SocketReceive client, \RawData := data;
9           SocketClose client;
10          Stop;
11      ENDPROC
12  ENDMODULE
```

*Figura 4: Código de um Cliente em RAPID*

conveniente.

Vamos entender o que cada linha desse código executa: VAR socketdev client -> Cria uma variável do tipo "socketdev"que irá ser o socket que irá se comunicar com um servidor.
VAR rawbytes data -> Cria uma variável do tipo "rawbytes"que irá armazenar informações recebidas vindas do servidor.
SocketCreate client -> Inicia um socket na variável criada previamente
SocketConnect client,"127.0.0.1",8080 -> Solicita a conexão entre o socket client e um servidor(outro socket) com o endereço IP em "127.0.0.1"na porta 8080.
SocketSend client, :="Teste-> Faz o socket client mandar uma string para o outro socket a qual já está previamente conectado
SocketReceive, := data -> Recebe uma resposta do servidor em formato de bytes. SocketClose client -> Encerra o socket criado.
Stop -> Encerra o programa

Mais informações sobre funções de WebSocket em RAPID disponiveis em [robotics 2010]
Importante ressaltar que o endereço IP deste exemplo é o endereço local na qual o software está sendo executado.Caso esse código esteja sendo simulado no RobotStudio irá funcionar, pois o controlador da simulação compartilha do endereço local com seu computador, isso se o server também estiver sendo executado no mesmo computador no endereço local.No caso de estiver utilizando o código no controlador real, o endereço utilizado terá que ser o endereço do seu computador na rede local entre computador e controlador.A porta "8080"é arbitraria e definida na criação do servidor descrita a seguir. Este código foi obtido em : [Jennings

```python
#!/usr/bin/env python3

import socket

HOST = '127.0.0.1'
PORT = 8080

with socket.socket(socket.AF_INET, socket.SOCK_STREAM) as s:
    s.bind((HOST, PORT))
    s.listen()
    conn, addr = s.accept()
    with conn:
        print('Connected by', addr)
        while True:
            data = conn.recv(1024)
            if not data:
                break
            conn.sendall(data)
```

*Figura 5: Código de um Server Simples em Python*

2019]. Neste simples código é realizado 4 coisas, com a mesma lógica usada no código do Cliente.Primeiro é criado um socket "s".Em seguida esse socket é atrelado a um endereço IP e uma porta definidos por "HOST"e "PORT"respectivamente (linha 9).
s.listen() apenas faz o programa aguardar algum outro socket tentar conectar ao server criado.Quando qualquer um tenta realizar a conexão o server aceita e armazena o socket do cliente em "conn"e "addr"representado o IP e a porta dele.
Por fim o server recebe algum dado em binário e o repassa para o client.Isso é feito nas linhas 15 e 18.

Quando utilizado no robô físico esse endereço local utilizado neste exemplo não será visto pelo controlador.O endereço a ser utilizado , normalmente, será "192.168.125.2"que por padrão é o endereço IP que o controlador fornece para a maquina conectada através da conexão ethernet.O número da porta é relativamente arbitrario, porém não será entrado em detalhes.Apenas escolha um número maior que 1023 para a porta.Lembre-se que o endereço e a porta devem coincidir em ambos cliente e server se a aplicação sendo feita for estática, ou seja nenhum desses valores mudar ao longo do tempo.

Se tudo ocorrer como o planejado estes dois programas devem interagir de forma simples entre eles.Porém em nenhum momento foi conectado à web.Para se realizar essa conexão, o server deve possuir métodos que realizem tal procedimento, uma vez que toda esta conexão demonstrada neste documento está localizada na rede local criada pelo controlador.Portanto seu computador deve estar conectado a rede local do controlador e também a internet e assim ser o intermediario entre o robô e a web.

# 4 Sugestões

Para evitar ligar o controlador a um computador inteiro, pode ser possível tentar se conectar através de um RaspBerry Pi, ou um roteador para efetuar a ponte entre a rede local e a Internet.

De acordo com a documentação da ABB, alguns controladores possuem uma porta WAN além das LAN para se realizar a conexão diretamente a rede Internet;Porém essa porta não foi encontrada pelo autor no robô disponível em laboratório.Porém isto é importante levar em conta ao se trabalhar com outras maquinas fora das localidades da faculdade.

Não foi realizado o teste para se conhecer como o controlador se comporta quando o mesmo realiza o papel de servidor.Assume-se que essa maneira se comporta de maneira analoga a mostrada neste documento, porém com os papeis invertidos.

Caso o IP não esteja sendo reconhecido durante a tentativa do Socket de se conectar, verifique o endereço IP de seu computador na rede local digitando "ipconfig"no prompt de comando.

## Referências

JENNINGS, N. **Socket Programming in Python (Guide)**. Disponível em: :https://realpython.com/python-sockets/: Real Python, 2019.

ROBOTICS, A. **Manual - RAPID Instructions, Functions and Data types**. Disponível em: https://drive.google.com/file/d/0B0z6VtggaV0kTTNxYnprNTBPR28/view: ABB, 2010.

ROBOTSTUDIO developercenter. **How to set up your PC to communicate with robot**. Disponível em: :http://developercenter.robotstudio.com/BlobProxy/devcenter/RobotCommunication/html/136d8d89-be7a-4c7d-9f50-ea9f11029aa7.htm: Robot Studio, 2019.