To control a TurtleBot3 robot from a remote computer without ROS installed, you can establish a network connection between the remote computer and the TurtleBot3 and send control commands using Python sockets. Here's an example of how you can achieve this:

On the TurtleBot3 (ROS side):

1. Create a new Python script, let's call it `turtlebot\_server.py`, and add the following code:

```python

import rospy

from geometry\_msgs.msg import Twist

from std\_msgs.msg import String

from socket import \*

def callback(data):

rospy.loginfo(rospy.get\_caller\_id() + 'Received command: %s', data.data)

# Parse the received command and control the robot accordingly

def turtlebot\_server():

rospy.init\_node('turtlebot\_server', anonymous=True)

rospy.Subscriber('cmd\_vel', Twist, callback)

rospy.spin()

if \_\_name\_\_ == '\_\_main\_\_':

turtlebot\_server()

```

2. Save the file and make it executable by running the following command in the terminal:

```bash

chmod +x turtlebot\_server.py

```

3. Run the script on the TurtleBot3:

```bash

rosrun <your\_package\_name> turtlebot\_server.py

```

Make sure to replace `<your\_package\_name>` with the actual package name where you saved the script.

On the remote computer (Python side):

1. Create a new Python script, let's call it `remote\_control.py`, and add the following code:

```python

from socket import \*

# IP address and port of the TurtleBot3

turtlebot\_ip = 'TURTLEBOT\_IP\_ADDRESS'

turtlebot\_port = 12345

# Connect to the TurtleBot3

client\_socket = socket(AF\_INET, SOCK\_STREAM)

client\_socket.connect((turtlebot\_ip, turtlebot\_port))

def send\_command(command):

client\_socket.send(command.encode())

# Example control commands

send\_command('forward')

send\_command('turn\_left')

send\_command('stop')

# Close the connection

client\_socket.close()

```

2. Replace `'TURTLEBOT\_IP\_ADDRESS'` with the actual IP address of the TurtleBot3. Make sure both the remote computer and the TurtleBot3 are on the same network.

3. Save the file and run the script on the remote computer:

```bash

python remote\_control.py

```

This script sends control commands to the TurtleBot3 by establishing a socket connection with the TurtleBot3's IP address and port, and then sending the commands as strings.

Note: The above code assumes that you have already set up ROS on the TurtleBot3 and have the necessary packages and dependencies installed.