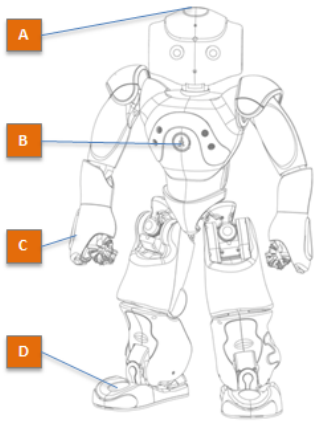
Tactility

MKI59: Robotlab practical
November 7, 2018

In this part we are going to work with the tactile sensors on the NAO.

1 Reading sensor values through ALMemory using ALModule

We will start today's lesson by walking through the Aldebaran example available at http://doc.aldebaran.com/2-1/naoqi/sensors/altouch.html tutorial. The Nao robot has several touch and tactile sensors; these are the colored areas on the Nao.



A: Head tactile sensor
B: Chest button
C: Tactile hands
D: Feet bumpers

Sensor values are stored in `ALMemory`. One way to read these sensor values is via the `getData()` method. A more convenient way to use sensors is to consider them as detectors of events. This holds for touch sensors (indicating whether the sensor has been touched or not), but also for collision detectors (near/far), for microphones capturing speech (word recognized or not) or cameras (face detected or not). Aldebaran has provided a convenient mechanism for its different sensors. You can `subscribe` to events on sensors by writing your own extension to the `ALModule` class of the specific module. All sensor modules can be found here: http://doc.aldebaran.com/2-1/naoqi/sensors/index.html#naoqi-sensors.

First we will make a class based on `ALTouch` called `ReactToTouch`, in which we specify what should be done upon detecting a particular memory change event from the touch sensors.

Code snippet 1: Example of subscribing to *TouchChanged* event

```
memory.subscribeToEvent("TouchChanged", "ReactToTouch", "onTouched")
```

For each sensor class, one or more events can be raised. The `ALTouch` generates the event `TouchChanged` to which your module can subscribe. Subscribing to an event adheres to well-known event-callback programming principles, as illustrated in the code snippet above.

Now, copy the following code, run it, and try to understand what happens. Please use a unique name where it says "yourname".

Code snippet 2: Example touch tutorial

```
# -*- encoding: UTF-8 -*-
""" Say 'My {Body_part} is touched' when receiving a touch event
"""

import time
import sys
import naoqi

from naoqi import ALModule
from naoqi import ALProxy
from naoqi import ALBroker

# Global variable to store the ReactToTouch module instance
ReactToTouch = None

ip = "192.168.1.102"
port = 9559

global tts
global memory
memory = ALProxy("ALMemory", ip, port)
tts = naoqi.ALProxy("ALTextToSpeech", ip, port)

class ReactToTouch(ALModule):
    """
    A simple module able to react to touch events.
    """
    def __init__(self, name):
        #We need to unsubscribe from any proxies still on the Nao to prevent errors
        try:
            p = ALProxy(name)
            p.exit()
        except:
            pass
        ALModule.__init__(self, name)
        # No need for IP and port here because
        # we have our Python broker connected to NAOqi broker

        # Create a proxy to ALTextToSpeech for later use
        self.tts = ALProxy("ALTextToSpeech")

        # Subscribe to TouchChanged event:
        memory.subscribeToEvent("TouchChanged", name, "onTouched")
        self.footTouched = False

    def onTouched(self, strVarName, value):
        """
        This will be called each time a touch is detected.
        """
        # Unsubscribe to the event when talking,
        # to avoid repetitions
        memory.unsubscribeToEvent("TouchChanged", "ReactToTouch")

        touched_bodies = []
```

```
        for p in value:
            if p[1]:
                touched_bodies.append(p[0])
        self.say(touched_bodies)

        # Subscribe again to the event
        memory.subscribeToEvent("TouchChanged", "ReactToTouch", "onTouched")

    def say(self, bodies):
        if (bodies == []):
            return

        sentence = "My " + bodies[0]
        for b in bodies[1:]:
            sentence = sentence + " and my " + b
        if (len(bodies) > 1):
            sentence = sentence + " are"
        else:
            sentence = sentence + " is"
        sentence = sentence + " touched."

        self.tts.say(sentence)

if __name__ == "__main__":
    pythonBroker = ALBroker("pythonBroker", "0.0.0.0", 9600, ip, port)
    yourname = ReactToTouch("yourname")
    try:
        while True:
            time.sleep(1)
    except KeyboardInterrupt:
        print
        print "Interrupted by user, shutting down"
        pythonBroker.shutdown()
        sys.exit(0)
```

The callback `onTouched()` is one of the two methods that are specified in the module `ReactToTouch` (the other being the initialization function). When a `TouchChanged` event happens, the method `onTouched` is called and the value of the touch event is given to this method. Subscribing and unsubscribing to events is further illustrated in the next example.