Laboratorio de Casas Inteligentes
Comunicación entre Procesos Usando un Blackboard
Practica No. 5

Objetivo: Usando un Blackboard interconectar procesos en C++ y Python que reciben y envian comandos a los sensores y actuadores, Internet of Things (IoT).

Desarrollo: Para cada uno de los siguientes apartados, realizar el software que se pide.
Duración: Dos semanas

1. El BlackBoard es una herramienta de paso de mensajes y repositorio de variables compartidas desarrollada en el laboratorio de Bio-Robótica por el MI. Mauricio Matamoros. Sirve para comunicar diferentes programas bajo los esquemas comando-respuesta y publicación suscripción. La manera en que es implementado es a través de un esquema cliente-servidor, donde el BlackBoard toma el rol de cliente y cada módulo o que se quiere conectar, se comporta como servidor. El BlackBoard utiliza un archivo de configuración XML para conocer la ubicación de cada módulo (direccion IP y puerto) para establecer una conexión y poder comunicarse con ellos. Cada módulo puede entonces enviar mensajes que corresponden a comandos, que deberán ser ejecutados por otro módulo, o bien, por el mismo BlackBoard, en caso de ser comandos estándar que el entienda. Se puede escribir a una variable compartida y todos los modulos que esten suscritos a ella recibirán una notificación de que el valor ha cambiado, para que puedan actuar en consecuencia si fuera necesario.
Para facilitar la implementación de nuevos modulos, se han desarrollado APIs de desarrollo que abstraen todos los detalles de conexión y manejo de mensajes de la aplicacion para la que fue disenada el modulo. Existen APIs para los lenguajes C#, C++, Python y Clips

Descargue en la página de las prácticas el archivo denominado Blackboard, el cual contiene los binarios del Blackboard, así como ejemplos en C++ y Python.

2. El código en C++ requiere de la herramienta de compilación BOOST, instalela utilizando el siguiente comando:

```
$ sudo apt-get install libboost-all-dev
```

Una vez instalado BOOST colocarse en el directorio: ~/Blackboard/example_arduino_C++

Edite CMakeLists.txt si es necesario cambiando los directorios fuentes y reconstruya el proyecto con cmake, si cmake no esta instalado, instalelo con:

```
$ sudo apt-get install cmake
```

Solamente la primera vez que se utilize cmake, primero hay que borrar el archivo CmakeCache.txt y el directorio Cmakefiles:

```
$ rm CmakeCache.txt
$ rm CMakeFiles
```

Después ejecute cmake:

```
$ cmake CMakeList.txt
```

Compile con make el código C/C++ sensor_actuator.cpp y monitor_house.cpp que se encuentran en ~/robots/Blackboard/example_arduino_C++/sensor_actuator_src:

```
$ make
```

Mono es una utilería de Linux que permite ejecutar programas desarrollados en el sistema operativo de Windows de Microsoft. Si no esta instalado en su sistema, instalelo para ejecutar el Blackboard.exe de la siguiente forma:

```
$ sudo apt-get install mono-complete
```

Una vez instalado ejecute el Blackboard.exe :

```
$ mono ~/Blackboard/BlackBoard_bin/Blackboard.exe
```

Cargue el archivo de configuración xml
~/Blackboard/BlackBoard_bin/BB_example_C++.xml

Estos dos pasos se pueden hacer directamente, estando en el directorio ~/Blackboard/BlackBoard_bin/ con:

```
$ mono Blackboard.exe BB_example_C++.xml
```

El archivo BB_example_C++.xml contiene la configuración de los programas que se interconectarán con el Blackboard, indicando las direcciones IPs, puertos y variables compartidas. El apendice A muestra este archivo. Inicialice el Blackboard con el boton de Start, el blackboard esperará que se conecten los procesos a través de el.

Blackboard utiliza librerías que se encuentran compiladas en el directorio ~/Blackboard/example_arduino_C++/uRobotics

Si los archivos compilados no funciona en su sistema Linux, por incompatibilidad del procesador, recompilelos de nuevo en la carpeta ~/Blackboard/librerias/uRobotics

```
$ rm CmakeCache.txt
$ rm CMakeFiles
$ cmake CMakeList.txt
$make
```

Después copie el directorio ~/Blackboard/librerias/uRobotics a ~/Blackboard/example_arduino_C++/uRobotics

En una terminal ejecute el código que se conecta por el puerto serie con el Arduino: ~/Blackboard/example_arduino_C++/bin/sensor_actuator

En otra terminal ejecute el código que monitorea los datos: ~/Blackboard/example_arduino_C++/bin/monitor_house

3. Para utilizar el código de Python copie el directorio ~/Blackboard/librerias/pyRobotics/pyrobotics/ al directorio /usr/lib/python2.7/pyrobotics o la directorio que contenga la versión de Python que esta utilizando.

El directorio pyrobotics/ contiene librerías de Python, desarrolladas por Adrían Revuelta, que se utilizan para establecer la comunicación con el Blackboard.

En una terminal ejecute el código que se conecta por el puerto serie con el Arduino utilizando Python:

```
$ python ~/Blackboard/example_python/sensor_actuator.py
```

En otra terminal ejecute el código que monitorea los datos con Python:

```
$ python ~/Blackboard/example_python/monitor_house_python.py
```

4.- Modifique el código que se ejecuta en el Arduino para que acepte, además de los comandos definidos en la práctica 2, los siguientes comandos, enviados por monitor_house o monitor_house_python y recibidos por sensor_actuator y sensor_actuator_python:

> Num.dispositivo shs continuous/stop all/temp/magnet/acelerometer/photo period(sec)
>
> ejem: A3 shs continuous all 10
>
> El dispositivo A3 enviará los valores de todos sensores cada 10 segundos.

APENDICE A

Configuración de los módulos que el Blackboard interconectará, formato en XML

```
<?xml version="1.0" encoding="UTF-8"?>
<blackboard version="1.0">

    <configuration>
        <name>BLK</name>
        <port>2300</port>
        <sendAttemps>2</sendAttemps>
        <commands />
        <startupSequence />
        <shutdownSequence />
    </configuration>

    <sharedVariables>
        <var name="sensor" type="string"/>
        <var name="actuator" type="string"/>
    </sharedVariables>

    <modules>

        <module name="MONITOR_HOUSE_C">
            <ip>127.0.0.1</ip>
            <ip>192.168.190.110</ip>
            <port>2025</port>
            <aliveCheck>true</aliveCheck>
            <commands>
                <command name="cmd_one" answer="True" parameters="True" timeout="2000" />
                <command name="cmd_two" answer="True" parameters="True" timeout="6000" />
            </commands>
        </module>


        <module name="SENSOR_ACTUATOR_C">
            <ip>127.0.0.1</ip>
            <ip>192.168.190.110</ip>
            <port>2011</port>
            <aliveCheck>true</aliveCheck>
            <commands>
                <command name="get_data" answer="True" parameters="True" timeout="1000" />
            </commands>
        </module>


        <module name="SENSOR_ACTUATOR_PYTHON">
            <ip>127.0.0.1</ip>
            <ip>192.168.190.110</ip>
            <port>2030</port>
            <aliveCheck>true</aliveCheck>
            <commands>
```

```
                <command name="cmd_one" answer="True" parameters="True" timeout="2000" />
                <command name="cmd_two" answer="True" parameters="True" timeout="6000" />
            </commands>
        </module>

        <module name="MONITOR_HOUSE_PYTHON">
            <ip>127.0.0.1</ip>
            <ip>192.168.190.110</ip>
            <port>2040</port>
            <aliveCheck>true</aliveCheck>
            <commands>
                <command name="cmd_one" answer="True" parameters="True" timeout="2000" />
                <command name="cmd_two" answer="True" parameters="True" timeout="6000" />
            </commands>
        </module>



    </modules>

</blackboard>
```