Cette annexe contient le code Java des deux activités de notre application :

-Activité 1 pour la sélection du robot (BluetoothPair.java)

-Activité 2 pour la connexion avec le robot et son contrôle (Control.java)

NB : Les codes présents dans le dossier sont commentés en français et ceux des annexes en anglais. En effet, lors de la création de ce code il a été créé et commenté en anglais. Seules les parties présentes dans le dossier ont été traduites.

Tout d'abord le code pour BluetoothPair.java :

```
package com.tyltcodeworld.louis.projetsade05;


import android.content.Intent;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.ListView;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.widget.TextView;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.Set;


public class BluetoothPair extends ActionBarActivity
{


    //widgets
    Button btnPaired;
    ListView BluetoothPair;
    //Bluetooth
    private BluetoothAdapter myBluetooth = null;
    private Set<BluetoothDevice> pairedDevices;
    public static String EXTRA_ADDRESS = "device_address";

    @Override
    protected void onCreate(Bundle savedInstanceState)
    {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bluetooth_pair);

        //Calling widgets
        btnPaired = (Button)findViewById(R.id.button);
        BluetoothPair = (ListView)findViewById(R.id.listView);

        //if the device has bluetooth
        myBluetooth = BluetoothAdapter.getDefaultAdapter();
```

```
        if(myBluetooth == null)
        {
            //Show a mensag. that the device has no bluetooth adapter
            Toast.makeText(getApplicationContext(), "Bluetooth Device Not
Available", Toast.LENGTH_LONG).show();

            //finish apk
            finish();
        }
        else if(!myBluetooth.isEnabled())
        {
            //Ask to the user turn the bluetooth on
            Intent turnBTon = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
            startActivityForResult(turnBTon,1);
        }

        btnPaired.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v)
            {
                pairedDevicesList();
            }
        });

    }



    private void pairedDevicesList()
    {
        pairedDevices = myBluetooth.getBondedDevices();
        ArrayList list = new ArrayList();

        if (pairedDevices.size()>0)
        {
            for(BluetoothDevice bt : pairedDevices)
            {
                list.add(bt.getName() + "\n" + bt.getAddress()); //Get the
device's name and the address
            }
        }
        else
        {
            Toast.makeText(getApplicationContext(), "No Paired Bluetooth
Devices Found.", Toast.LENGTH_LONG).show();
        }

        final ArrayAdapter adapter = new
ArrayAdapter(this,android.R.layout.simple_list_item_1, list);
        BluetoothPair.setAdapter(adapter);
        BluetoothPair.setOnItemClickListener(myListClickListener); //Method
called when the device from the list is clicked

    }
```

```
    private AdapterView.OnItemClickListener myListClickListener = new
AdapterView.OnItemClickListener()
    {
        public void onItemClick (AdapterView<?> av, View v, int arg2, long
arg3)
        {
            // Get the device MAC address, the last 17 chars in the View
            String info = ((TextView) v).getText().toString();
            String address = info.substring(info.length() - 17);

            // Make an intent to start next activity.
            Intent i = new Intent(BluetoothPair.this, Control.class);

            //Change the activity.
            i.putExtra(EXTRA_ADDRESS, address); //this will be received at
Control (class) Activity
            startActivity(i);
        }
    };


    @Override
    public boolean onCreateOptionsMenu(Menu menu)
    {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.menu_bluetooth_pair, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

Et maintenant le code pour Control.java

```
package com.tyltcodeworld.louis.projetsade05;


import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
```

```
import android.bluetooth.BluetoothSocket;
import android.content.Intent;
import android.view.View;
import android.widget.Button;
import android.widget.CompoundButton;
import android.widget.CompoundButton.OnCheckedChangeListener;
import android.widget.SeekBar;
import android.widget.TextView;
import android.widget.Switch;
import android.widget.CheckBox;
import android.widget.Toast;
import android.app.ProgressDialog;
import android.bluetooth.BluetoothAdapter;
import android.bluetooth.BluetoothDevice;
import android.os.AsyncTask;

import java.io.IOException;
import java.util.UUID;


public class Control extends ActionBarActivity {

    Button btnDis, btnlup, btnup, btnrup, btnleft, btnstop, btnright, btnlown,
btndown, btnrown;
    CheckBox cboxactiv;
    String address = null;
    private ProgressDialog progress;
    BluetoothAdapter myBluetooth = null;
    BluetoothSocket btSocket = null;
    private boolean isBtConnected = false;
    //SPP UUID. Look for it
    static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-
00805F9B34FB");
    private Switch Switcher;


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        Intent newint = getIntent();
        address = newint.getStringExtra(BluetoothPair.EXTRA_ADDRESS);
//receive the address of the bluetooth Pair

        //view of the lControl
        setContentView(R.layout.activity_control);

        //call the widgtes
        btnDis = (Button) findViewById(R.id.button4);
        btnlup = (Button) findViewById(R.id.lup);
        btnup = (Button) findViewById(R.id.up);
        btnrup = (Button) findViewById(R.id.rup);
        btnleft = (Button) findViewById(R.id.left);
        btnstop = (Button) findViewById(R.id.stop);
        btnright = (Button) findViewById(R.id.right);
        btnlown = (Button) findViewById(R.id.lown);
        btndown = (Button) findViewById(R.id.down);
```

```
        btnrown = (Button) findViewById(R.id.rown);



        cboxactiv = (CheckBox) findViewById(R.id.Switcher1);


        new ConnectBT().execute(); //Call the class to connect

        //commands to be sent to bluetooth


        btnDis.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Disconnect(); //close connection
            }
        });

        btnlup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                lup();
            }
        });
        btnup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                up();
            }
        });
        btnrup.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                rup();
            }
        });

        btnleft.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                left();
            }
        });
        btnstop.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                stop();
            }
        });
        btnright.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                right();
            }
        });
```

```
        btnlown.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                lown();
            }
        });
        btndown.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                down();
            }
        });
        btnrown.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                rown();
            }
        });


        cboxactiv.setOnClickListener(new View.OnClickListener() {
            public void onClick(View v) {
                    if (((CheckBox) v).isChecked()) {
                        if (btSocket != null) {
                            try {

btSocket.getOutputStream().write("o".toString().getBytes());
                            } catch (IOException e) {
                                msg("Error");
                            }
                        }
                    }

                    else {
                        if (btSocket != null) {
                            try {

btSocket.getOutputStream().write("f".toString().getBytes());
                            } catch (IOException e) {
                                msg("Error");
                            }
                        }
                    }
            }
        });


    }



    private void Disconnect()
    {
        if (btSocket!=null) //If the btSocket is busy
        {
            try
```

```
            {
                btSocket.close(); //close connection
            }
            catch (IOException e)
            { msg("Error");}
        }
        finish(); //return to the first layout

    }

    private void lup()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("e".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void up()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("z".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void rup()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("r".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void left()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("q".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void stop()
    {
```

```
        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("a".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void right()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("d".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void lown()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("w".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }
    }
    private void down()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("s".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }

        }
    }
    private void rown()
    {

        if (btSocket!=null) {
            try {
                btSocket.getOutputStream().write("c".toString().getBytes());
            } catch (IOException e) {
                msg("Error");
            }
        }

    }
```

```
    // fast way to call Toast
    private void msg(String s)
    {
        Toast.makeText(getApplicationContext(),s,Toast.LENGTH_LONG).show();
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
present.
        getMenuInflater().inflate(R.menu.menu_control, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }

    private class ConnectBT extends AsyncTask<Void, Void, Void>  // UI thread
    {
        private boolean ConnectSuccess = true; //if it's here, it's almost
connected

        @Override
        protected void onPreExecute()
        {
            progress = ProgressDialog.show(Control.this, "Connecting...",
"Please wait!!!");  //show a progress dialog
        }

        @Override
        protected Void doInBackground(Void... devices) //while the progress
dialog is shown, the connection is done in background
        {
            try
            {
                if (btSocket == null || !isBtConnected)
```

```
                {
                    myBluetooth = BluetoothAdapter.getDefaultAdapter();//get
the mobile bluetooth device
                    BluetoothDevice dispositivo =
myBluetooth.getRemoteDevice(address);//connects to the device's address and
checks if it's available
                    btSocket =
dispositivo.createInsecureRfcommSocketToServiceRecord(myUUID);//create a
RFCOMM (SPP) connection
                    BluetoothAdapter.getDefaultAdapter().cancelDiscovery();
                    btSocket.connect();//start connection
                }
            }
            catch (IOException e)
            {
                ConnectSuccess = false;//if the try failed, you can check the
exception here
            }
            return null;
        }
        @Override
        protected void onPostExecute(Void result) //after the doInBackground,
it checks if everything went fine
        {
            super.onPostExecute(result);

            if (!ConnectSuccess)
            {
                msg("Connection Failed. Is it a SPP Bluetooth? Try again.");
                finish();
            }
            else
            {
                msg("Connected.");
                isBtConnected = true;
            }
            progress.dismiss();
        }
    }
}
```