

Annexe DB : Form2.cpp

Annexe DB : Form2.cpp

```
#include "StdAfx.h"
#include "Form2.h"

namespace Cmd_pic {

System::Void Form2::Form2_Load(System::Object^ sender, System::EventArgs^ e) //Fonction s'exécutant à l'ouverture de la form2
{
    int i = 0;
    while (i < 16) //Permet d'avoir une valeur dans le fichier de config même si tous les champs ne sont pas remplis
    {
        m_cfgservo1[i] = "50000";
        m_cfgservo2[i] = "50000";
        i++;
    }
    if (File::Exists("config_moteur.txt") && File::Exists("config_sliders.txt")) //Lecture des fichiers de configuration
    {
        StreamReader ^sr3;
        String ^line3;
        StreamReader ^sr4;
        String ^line4;
        int k = 0;
        int l = 0;
        sr3 = gcnew StreamReader("config_moteur.txt");
        line3 = sr3->ReadLine();
        while (line3 != nullptr)
        {
            m_cfmoteur[k] = line3;
            line3 = sr3->ReadLine();
            k++;
        }
        sr4 = gcnew StreamReader("config_sliders.txt");
        line4 = sr4->ReadLine();
        while (line4 != nullptr)
        {
            m_cfgslider[l] = line4;
            line4 = sr4->ReadLine();
            l++;
        }
        sr3->Close();
        sr4->Close();

        trackBar_roue1->Minimum = Convert::ToInt32(m_cfgslider[0]);
        textBox_valS1min->Text = m_cfgslider[0];

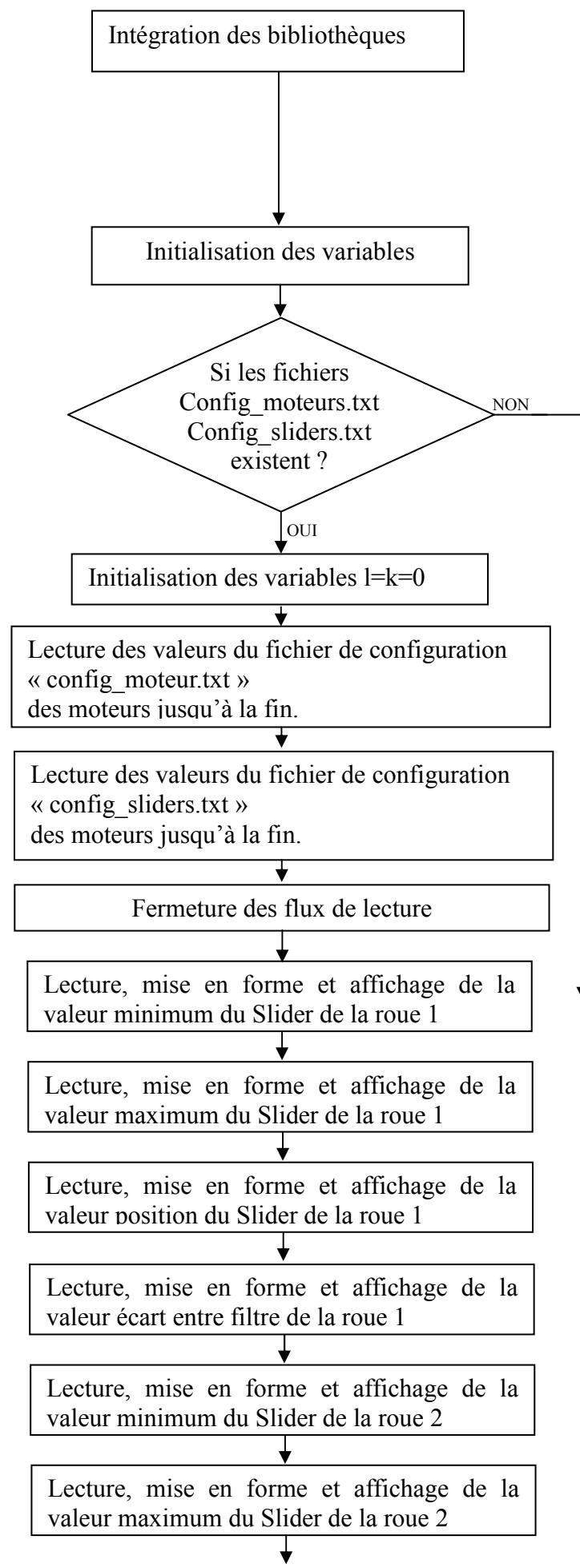
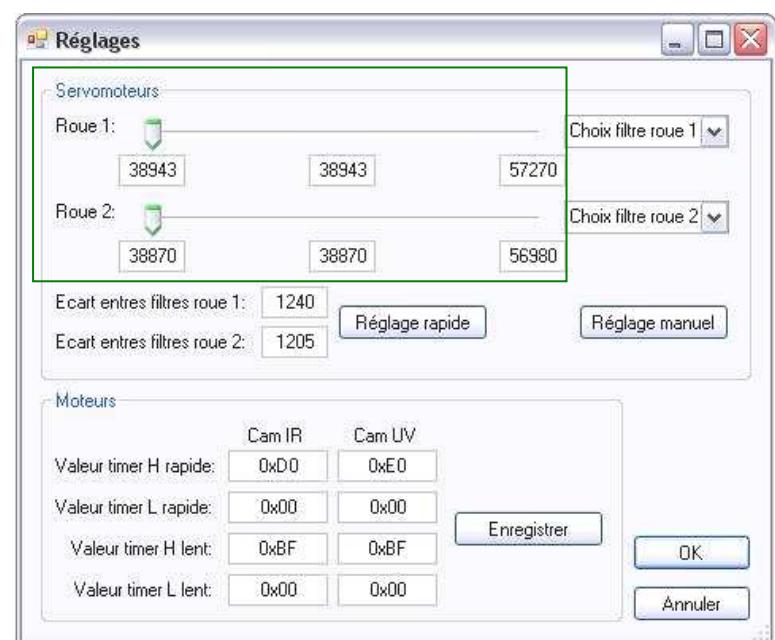
        trackBar_roue1->Maximum = Convert::ToInt32(m_cfgslider[1]);
        textBox_valS1max->Text = m_cfgslider[1];

        trackBar_roue1->Value = Convert::ToInt32(m_cfgslider[2]);
        textBox_valSlider1->Text = m_cfgslider[2];

        m_ecart1 = Convert::ToInt32(m_cfgslider[3]);
        textBox_ecart1->Text = m_cfgslider[3];

        trackBar_roue2->Minimum = Convert::ToInt32(m_cfgslider[4]);
        textBox_valS2min->Text = m_cfgslider[4];

        trackBar_roue2->Maximum = Convert::ToInt32(m_cfgslider[5]);
        textBox_valS2max->Text = m_cfgslider[5];
    }
}
}
```



Annexe DB : Form2.cpp

```

trackBar_roue2->Value = Convert::ToInt32(m_cfgslider[6]);
textBox_valslider2->Text = m_cfgslider[6];

m_ecart2 = Convert::ToInt32(m_cfgslider[7]);
textBox_ecart2->Text = m_cfgslider[7];

textBox_tmrHfast->Text = m_cfgmoteur[0];
textBox_tmrLfast->Text = m_cfgmoteur[1];

textBox_tmrHlow->Text = m_cfgmoteur[2];
textBox_tmrLlow->Text = m_cfgmoteur[3];
}

}

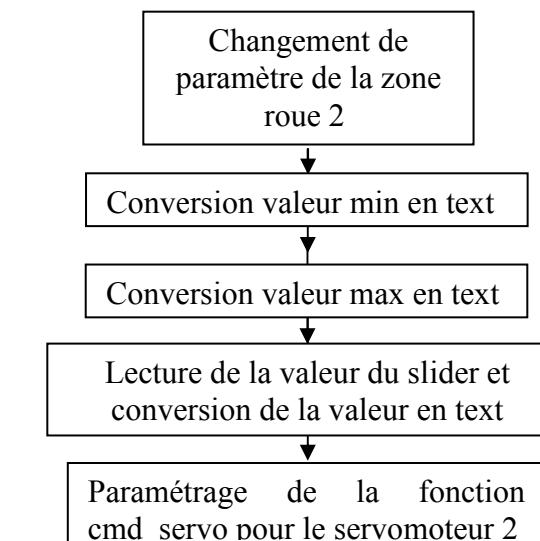
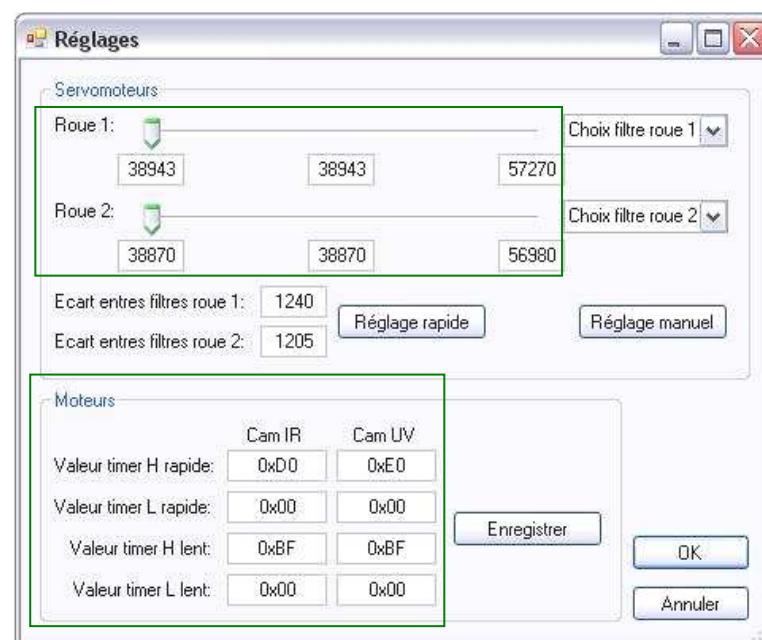
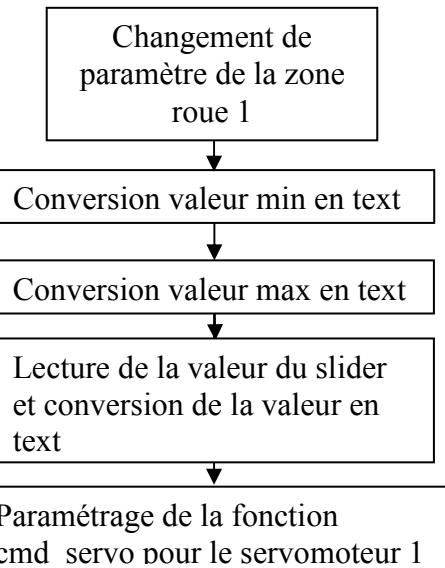
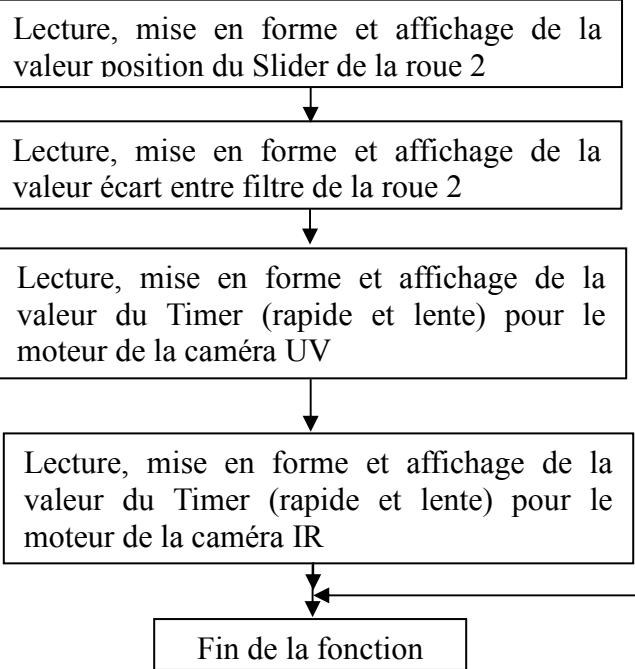
```

```

System::Void Form2::trackBar_roue1_ValueChanged(System::Object ^sender, System::EventArgs ^e)
{
    trackBar_roue1->Minimum = Convert::ToInt32(textBox_valS1min->Text);
    trackBar_roue1->Maximum = Convert::ToInt32(textBox_valS1max->Text);
    textBox_valslider1->Text = trackBar_roue1->Value.ToString();
    frm1->cmd_servo(trackBar_roue1->Value, 0);
}

System::Void Form2::trackBar_roue2_ValueChanged(System::Object^ sender, System::EventArgs^ e)
{
    trackBar_roue2->Minimum = Convert::ToInt32(textBox_valS2min->Text);
    trackBar_roue2->Maximum = Convert::ToInt32(textBox_valS2max->Text);
    textBox_valslider2->Text = trackBar_roue2->Value.ToString();
    frm1->cmd_servo(trackBar_roue2->Value, 1);
}

```



Annexe DB : Form2.cpp

```
System::Void Form2::comboBox_roue1_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)
{
    m_cfgservo1[comboBox_roue1->SelectedIndex] = trackBar_roue1->Value.ToString();
}

System::Void Form2::comboBox_roue2_SelectedIndexChanged(System::Object^ sender, System::EventArgs^ e)
{
    m_cfgservo2[comboBox_roue2->SelectedIndex] = trackBar_roue2->Value.ToString();
}
```

Enregistrement de la valeur du filtre en cours (lors du paramétrage en mode manuel de la roue1)

Enregistrement de la valeur du filtre en cours (lors du paramétrage en mode manuel de la roue2)

```
System::Void Form2::button_rec_Click(System::Object^ sender, System::EventArgs^ e) //Bouton enregistrement manuel (on défini chaque filtres un par un)
{
    StreamWriter ^sw1;
    StreamWriter ^sw2;
    StreamWriter ^sw3;
    sw1 = gcnew StreamWriter("config_roue1.txt");
    sw2 = gcnew StreamWriter("config_roue2.txt");
    sw3 = gcnew StreamWriter("config_sliders.txt");
    for each (String ^ s in m_cfgservo1) sw1->WriteLine(s);
    for each (String ^ s in m_cfgservo2) sw2->WriteLine(s);
    trackBar_roue1->Value = trackBar_roue1->Minimum;
    trackBar_roue2->Value = trackBar_roue2->Minimum;
    m_cfgslider[0] = trackBar_roue1->Minimum.ToString();
    m_cfgslider[1] = trackBar_roue1->Maximum.ToString();
    m_cfgslider[2] = trackBar_roue1->Value.ToString();
    m_cfgslider[3] = m_ecart1.ToString();
    m_cfgslider[4] = trackBar_roue2->Minimum.ToString();
    m_cfgslider[5] = trackBar_roue2->Maximum.ToString();
    m_cfgslider[6] = trackBar_roue2->Value.ToString();
    m_cfgslider[7] = m_ecart2.ToString();
    for each (String ^ s in m_cfgslider) sw3->WriteLine(s);
    sw1->Close();
    sw2->Close();
    sw3->Close();
}
```

Création des flux d'écriture SW1 à SW3 avec :
 SW1 pour config_roue1.txt
 SW2 pour config_roue2.txt
 SW3 pour config_sliders.txt

Pour chaque position de filtres de la roue1, sauvegarder la position + Ecriture dans le fichier config_roue1.txt correspondant au flux SW1.

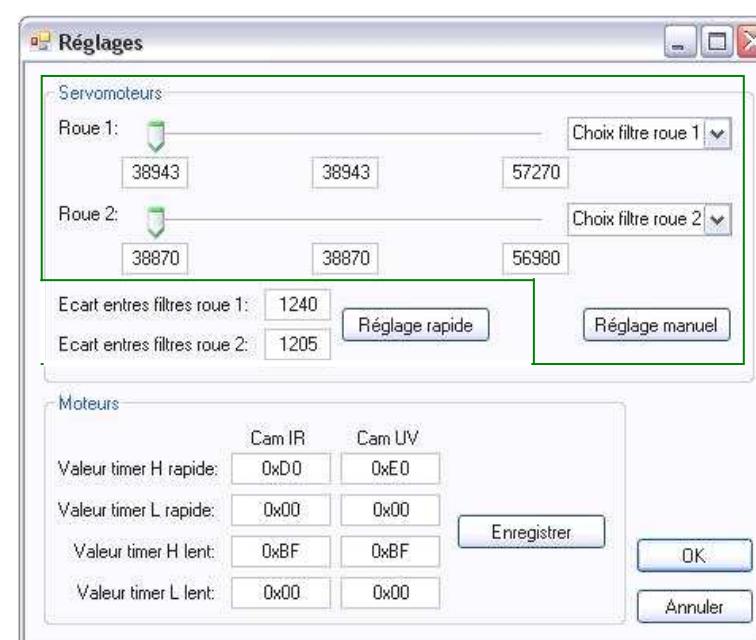
Pour chaque position de filtres de la roue2, sauvegarder la position + Ecriture dans le fichier config_roue2.txt correspondant au flux SW2.

Sauvegarder des valeurs minimum, maximum, position et écart du slider de la roue1

Sauvegarder des valeurs minimum, maximum, position et écart du slider de la roue1

Ecriture dans le fichier config_sliders.txt config_roue1.txt correspondant au flux SW3.

Fermeture des flux SW1 à SW3

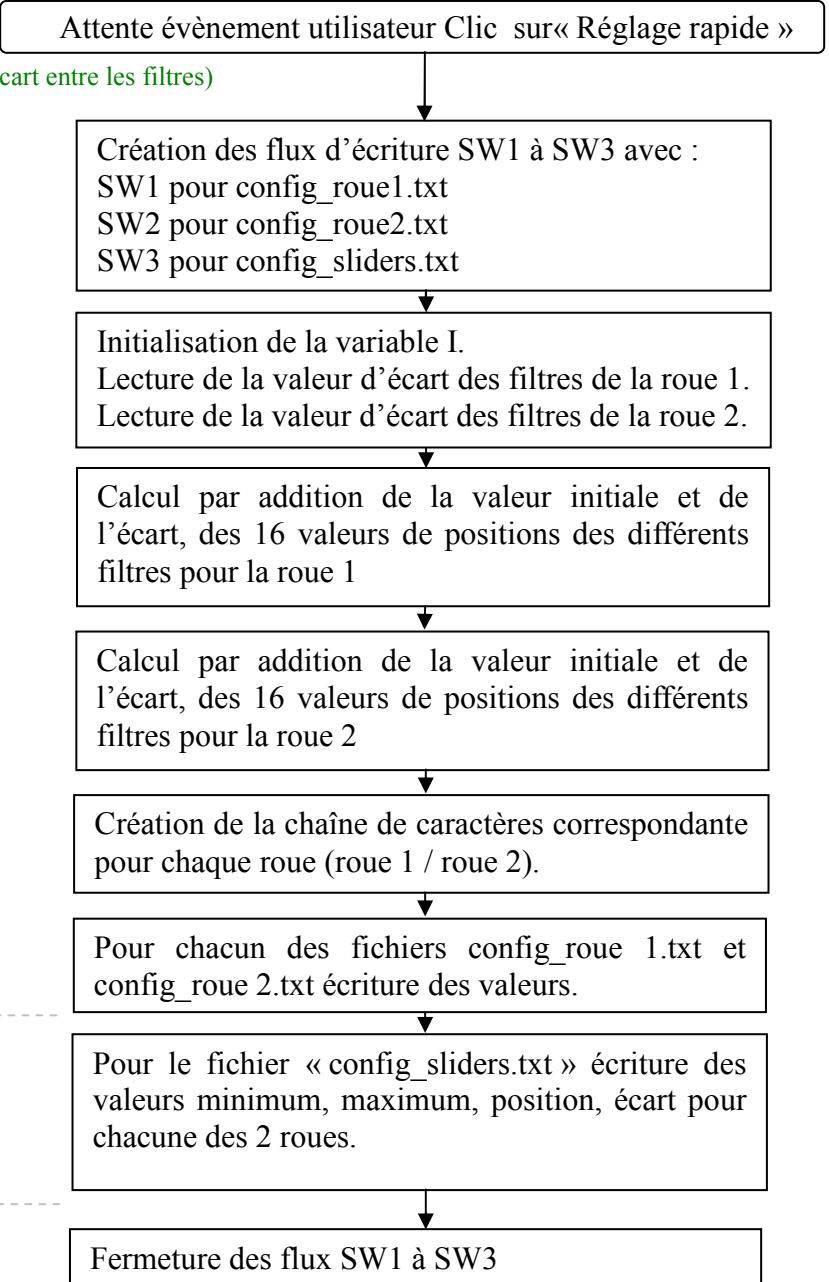


Annexe DB : Form2.cpp

```

System::Void Form2::button_recfast_Click(System::Object^ sender, System::EventArgs^ e) //Bouton enregistrement rapide (on définit les filtres 1 des 2 roues ainsi que l'écart entre les filtres)
{
    StreamWriter ^sw1;
    StreamWriter ^sw2;
    StreamWriter ^sw3;
    int i = 0;
    m_ecart1 = Convert::ToInt32(textBox_ecart1->Text);
    m_ecart2 = Convert::ToInt32(textBox_ecart2->Text);
    for (i = 1; i<16; i++)
    {
        int a = Convert::ToInt32(m_cfgservo1[i-1]);
        a = a + m_ecart1;
        int b = Convert::ToInt32(m_cfgservo2[i-1]);
        b = b + m_ecart2;
        m_cfgservo1[i] = a.ToString();
        m_cfgservo2[i] = b.ToString();
    }
    sw1 = gcnew StreamWriter("config_roue1.txt");
    sw2 = gcnew StreamWriter("config_roue2.txt");
    sw3 = gcnew StreamWriter("config_sliders.txt");
    for each (String ^s in m_cfgservo1) sw1->WriteLine(s);
    for each (String ^s in m_cfgservo2) sw2->WriteLine(s);
    trackBar_roue1->Value = trackBar_roue1->Minimum;
    trackBar_roue2->Value = trackBar_roue2->Minimum;
    m_cfgslider[0] = trackBar_roue1->Minimum.ToString();
    m_cfgslider[1] = trackBar_roue1->Maximum.ToString();
    m_cfgslider[2] = trackBar_roue1->Value.ToString();
    m_cfgslider[3] = m_ecart1.ToString();
    m_cfgslider[4] = trackBar_roue2->Minimum.ToString();
    m_cfgslider[5] = trackBar_roue2->Maximum.ToString();
    m_cfgslider[6] = trackBar_roue2->Value.ToString();
    m_cfgslider[7] = m_ecart2.ToString();
    for each (String ^s in m_cfgslider) sw3->WriteLine(s);
    sw1->Close();
    sw2->Close();
    sw3->Close();
}

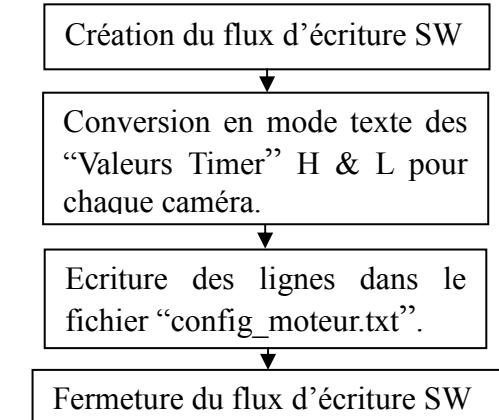
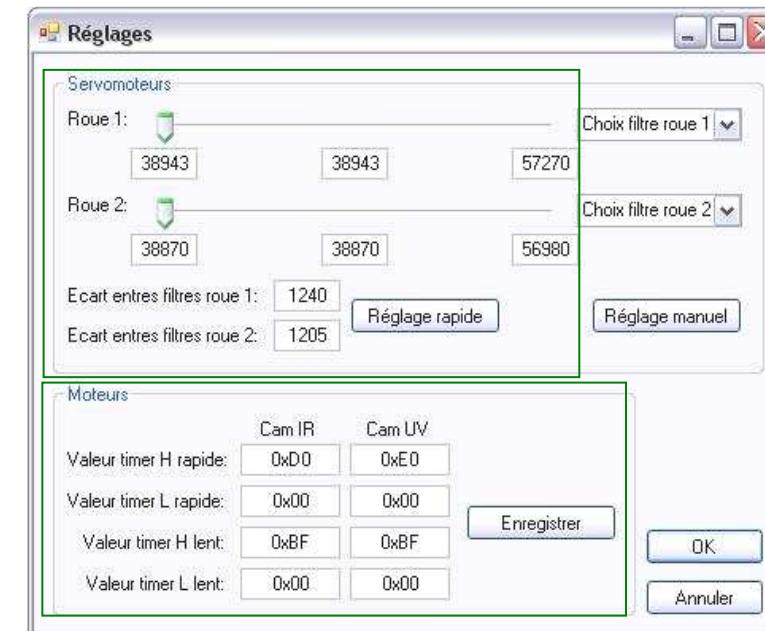
```



```

System::Void Form2::button_recmoteur_Click(System::Object^ sender, System::EventArgs^ e) //Enregistrement des valeurs de vitesse moteur
{
    StreamWriter ^sw;
    sw = gcnew StreamWriter("config_moteur.txt");
    m_cfgmoteur[0] = textBox_tmrHfast->Text;
    m_cfgmoteur[1] = textBox_tmrLfast->Text;
    m_cfgmoteur[2] = textBox_tmrHlow->Text;
    m_cfgmoteur[3] = textBox_tmrLlow->Text;
    m_cfgmoteur[4] = textBox_tmrHfastUV->Text;
    m_cfgmoteur[5] = textBox_tmrLfastUV->Text;
    m_cfgmoteur[6] = textBox_tmrHlowUV->Text;
    m_cfgmoteur[7] = textBox_tmrLlowUV->Text;
    for each (String ^s in m_cfgmoteur) sw->WriteLine(s);
    sw->Close();
}

```

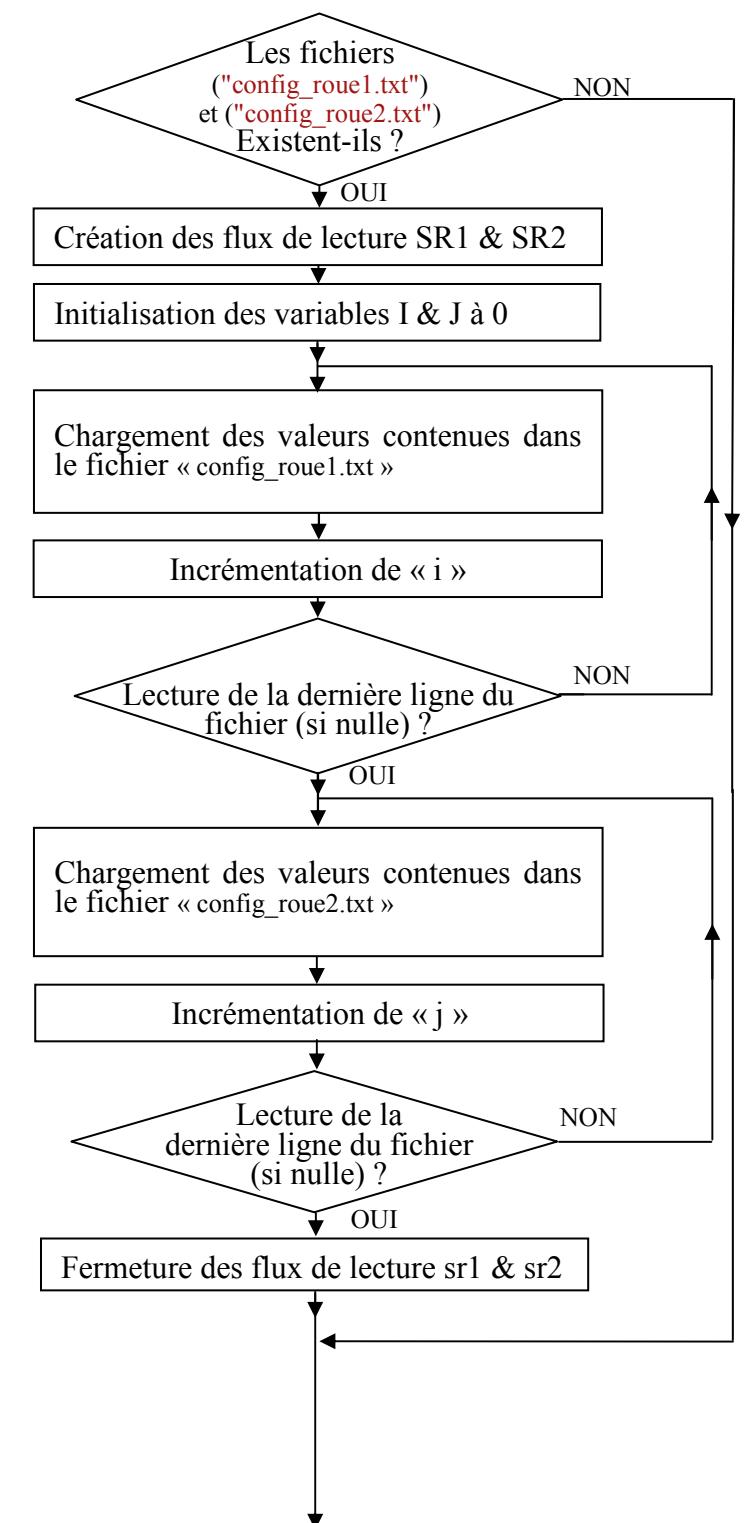
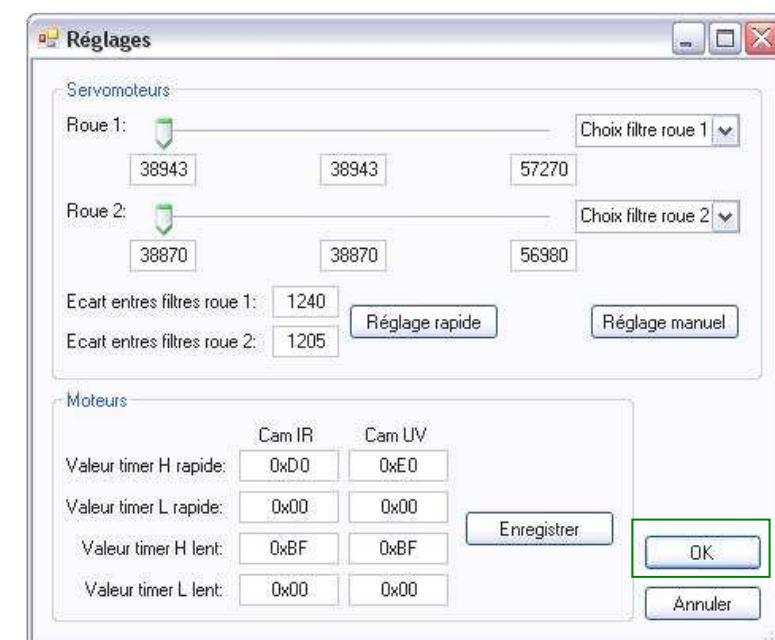


Annexe DB : Form2.cpp

```

System::Void Form2::button_ok_Click(System::Object^ sender, System::EventArgs^ e) //Lors du clic sur OK mise à jour des valeurs dans la Form1
{
    if (File::Exists("config_roue1.txt") && File::Exists("config_roue2.txt"))
    {
        StreamReader ^sr1;
        String ^line1;
        StreamReader ^sr2;
        String ^line2;
        int i = 0;
        int j = 0;
        sr1 = gcnew StreamReader("config_roue1.txt");
        line1 = sr1->ReadLine();
        while (line1 != nullptr)
        {
            frm1->m_cfgroue1[i] = line1;
            line1 = sr1->ReadLine();
            i++;
        }
        sr1->Close();
        sr2 = gcnew StreamReader("config_roue2.txt");
        line2 = sr2->ReadLine();
        while (line2 != nullptr)
        {
            frm1->m_cfgroue2[j] = line2;
            line2 = sr2->ReadLine();
            j++;
        }
        sr2->Close();
    }
}

```



Annexe DB : Form2.cpp

```

if(File::Exists("config_moteur.txt")&& File::Exists("config_sliders.txt"))
{
    StreamReader ^sr3;
    String ^line3;
    StreamReader ^sr4;
    String ^line4;
    int k = 0;
    int l = 0;
    sr3 = gcnew StreamReader("config_moteur.txt");
    line3 = sr3->ReadLine();
    while (line3 != nullptr)
    {
        m_cfgmoteur[k] = line3;
        line3 = sr3->ReadLine();
        k++;
    }

    sr4 = gcnew StreamReader("config_sliders.txt");
    line4 = sr4->ReadLine();
    while (line4 != nullptr)
    {
        m_cfgslider[l] = line4;
        line4 = sr4->ReadLine();
        l++;
    }
    sr3->Close();
    sr4->Close();
}

frm1->m_tmrH1 = Convert::ToInt32(m_cfgmoteur[0], 16);
frm1->m_tmrL1 = Convert::ToInt32(m_cfgmoteur[1], 16);
frm1->m_tmrH2 = Convert::ToInt32(m_cfgmoteur[2], 16);
frm1->m_tmrL2 = Convert::ToInt32(m_cfgmoteur[3], 16);
frm1->m_timerUVH1 = Convert::ToInt32(m_cfgmoteur[4], 16);
frm1->m_timerUVL1 = Convert::ToInt32(m_cfgmoteur[5], 16);
frm1->m_timerUVH2 = Convert::ToInt32(m_cfgmoteur[6], 16);
frm1->m_timerUVL2 = Convert::ToInt32(m_cfgmoteur[7], 16);

}
Form2::Close();
}

```

