```cpp
#include "EC12P.h"

namespace Hardware
{
    /*! Constructor*/
    EC12P::EC12P()
    {
        // Init Rotary button
        Button.SetDirection(GPIO::Input);
        Button.SetEdge(GPIO::Rising);

        // Init Encoder
        Rotary.set_period(100000000L);

        // Init Encoder color
        R.SetDirection(GPIO::Output);
        B.SetDirection(GPIO::Output);
        G.SetDirection(GPIO::Output);
        SetPixelColor(None);

        threadRunning = false;
    }

    /*! De-constructor*/
    EC12P::~EC12P() {   }

    /*! Set the shaft color
    \param value as Color enumerator
    */
    void EC12P::SetPixelColor(Color value)
    {
        switch (value)
        {
        case Hardware::EC12P::Red:
            R.SetValue(GPIO::High);
            B.SetValue(GPIO::Low);
            G.SetValue(GPIO::Low);
            break;
        case Hardware::EC12P::Pink:
            R.SetValue(GPIO::High);
            B.SetValue(GPIO::High);
        G.SetValue(GPIO::Low);
        break;
    case Hardware::EC12P::Blue:
        R.SetValue(GPIO::Low);
        B.SetValue(GPIO::High);
        G.SetValue(GPIO::Low);
        break;
    case Hardware::EC12P::SkyBlue:
        R.SetValue(GPIO::Low);
        B.SetValue(GPIO::High);
        G.SetValue(GPIO::High);
        break;
    case Hardware::EC12P::Green:
        R.SetValue(GPIO::Low);
        B.SetValue(GPIO::Low);
        G.SetValue(GPIO::High);
        break;
    case Hardware::EC12P::Yellow:
        R.SetValue(GPIO::High);
        B.SetValue(GPIO::Low);
        G.SetValue(GPIO::High);
        break;
    case Hardware::EC12P::White:
        R.SetValue(GPIO::High);
        B.SetValue(GPIO::High);
        G.SetValue(GPIO::High);
        break;
    case Hardware::EC12P::None:
        R.SetValue(GPIO::Low);
        B.SetValue(GPIO::Low);
        G.SetValue(GPIO::Low);
        break;
    }
    PixelColor = value;
}

/*! Loops through all the colors except of as a thread  */
void EC12P::RainbowLoop(int sleepperiod)
{
    this->sleepperiod = sleepperiod;
```

```
        this->threadRunning = true;
        if (pthread_create(&thread, NULL, colorLoop, this)) {   throw Exception::FailedToCreateThreadException();   }
    }

    /*! The thread function that runs trough all the colors*/
    void *colorLoop(void *value)
    {
        int i = 0;
        EC12P *ec12p = static_cast<EC12P*>(value);
        EC12P::Color pcolor;
        while (ec12p->threadRunning)
        {
            pcolor = static_cast<EC12P::Color>(i);
            ec12p->SetPixelColor(pcolor);
            usleep(ec12p->sleepperiod);
            i++;
            if (i == 6) { i = 0; }
        }
        return ec12p;
    }
}
```