```cpp
/*
* This code is based upon:
* Derek Molloy, "Exploring BeagleBone: Tools and Techniques for Building
* with Embedded Linux", Wiley, 2014, ISBN:9781118935125.
* See: www.exploringbeaglebone.com
*/

#pragma once
#include "BBB.h"

#define EXPORT_PIN "/sys/class/gpio/export"
#define UNEXPORT_PIN "/sys/class/gpio/unexport"
#define GPIOS "/sys/class/gpio/gpio"
#define DIRECTION "/direction"
#define VALUE "/value"
#define EDGE "/edge"

using namespace std;

namespace Hardware
{
    class GPIO:
        public BBB
    {
    public:
        enum Direction{ Input, Output };
        enum Value{ Low = 0, High = 1 };
        enum Edge{ None, Rising, Falling, Both };

        int number; // Number of the pin

        int WaitForEdge();
        int WaitForEdge(CallbackType callback);
        void WaitForEdgeCancel() { this->threadRunning = false; }

        Value GetValue();
        void SetValue(Value value);

        Direction GetDirection();
        void SetDirection(Direction direction);

        Edge GetEdge();
        void SetEdge(Edge edge);

        GPIO(int number);
        ~GPIO();

    private:
        string gpiopath;
        Direction direction;
        Edge edge;
        friend void* threadedPollGPIO(void *value);

        bool isExported(int number, Direction &dir, Edge &edge);
        bool ExportPin(int number);
        bool UnexportPin(int number);

        Direction ReadsDirection(const string &gpiopath);
        void WritesDirection(const string &gpiopath, Direction direction);

        Edge ReadsEdge(const string &gpiopath);
        void WritesEdge(const string &gpiopath, Edge edge);

        Value ReadsValue(const string &gpiopath);
        void WritesValue(const string &gpiopath, Value value);
    };

    void* threadedPollGPIO(void *value);
}
```