Private Entrepreneur (PE) Kabatskiy Vladimir Viktorovich, Novosibirsk, 2021

Device&Tools:

ETH signal generator

Introduce: periodic signal generator with Ethernet data transmission and control interfaces

Components:

PC software

Features of the signal generator software:

Java 8 SE only

Features of the signal generator:

1. Generating test signals (sinus wave, triangle wave, meander wave) with variable amplitude and frequency
2. Transmitting data and commands in separate threads
3. The signal generator is automatically connected to the signal receiver. It is enough just to specify the IP address of the signal receiver
4. The signal generator is controlled by the signal receiver
5. Can be restarted from the console

Fragment of the program code:

```
import java.io.IOException;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.TimeUnit;
import java.util.logging.Level;
import java.util.logging.Logger;

/**
 *
 * @author Copyright (c) 2021 IE Vladimir Kabatzcky
 */
public class Main {

    private static final int DATA_SERVICE_PERIOD = 10;//10(ms)

    private static DataHandlingThread mDataThread;
    private static ScheduledExecutorService mDataService;

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        CmdHandlingThread cmdThread = new CmdHandlingThread();
        cmdThread.start();

        mDataThread = new DataHandlingThread();
        mDataService = Executors
                .newSingleThreadScheduledExecutor();
        mDataService.scheduleAtFixedRate(mDataThread, 0, DATA_SERVICE_PERIOD,
TimeUnit.MILLISECONDS);

        cmdThread.setDataHandlingThread(mDataThread);
```

```
            while (true) {
                try {
                    int code = System.in.read();
                    char ch = (char) code;
                    System.out.println("ch: " + ch + ", code: " + code);
                    switch (ch) {
                        case 'x':
                            cmdThread.abort();
                            while (cmdThread.isAlive()) {
                                try {
                                    TimeUnit.MILLISECONDS.sleep(50);
                                } catch (InterruptedException ex) {
                                    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
                                }
                            }
                            if (mDataService != null) {
                                mDataService.shutdown();
                            }
                            System.exit(0);
                            break;
                        case 'r':
                            cmdThread.reset();
                            break;
                    }
                    try {
                        TimeUnit.MILLISECONDS.sleep(100);
                    } catch (InterruptedException ex) {
                        Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
                    }
                } catch (IOException ex) {
                    Logger.getLogger(Main.class.getName()).log(Level.SEVERE, null, ex);
                }
            }
        }
    }
```

Thanks for attention!