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

Device&Tools:

Mobile oscilloscope client

Introduce: mobile client for visualizing data received from an analog-to-digital voltage converter via a WiFi interface

Components:

1. Mobile client software (visualizer)

Features of the visualizer software:

1. Java 8 SE
2. Android 7.0 (Nougat)

Features of the visualizer:

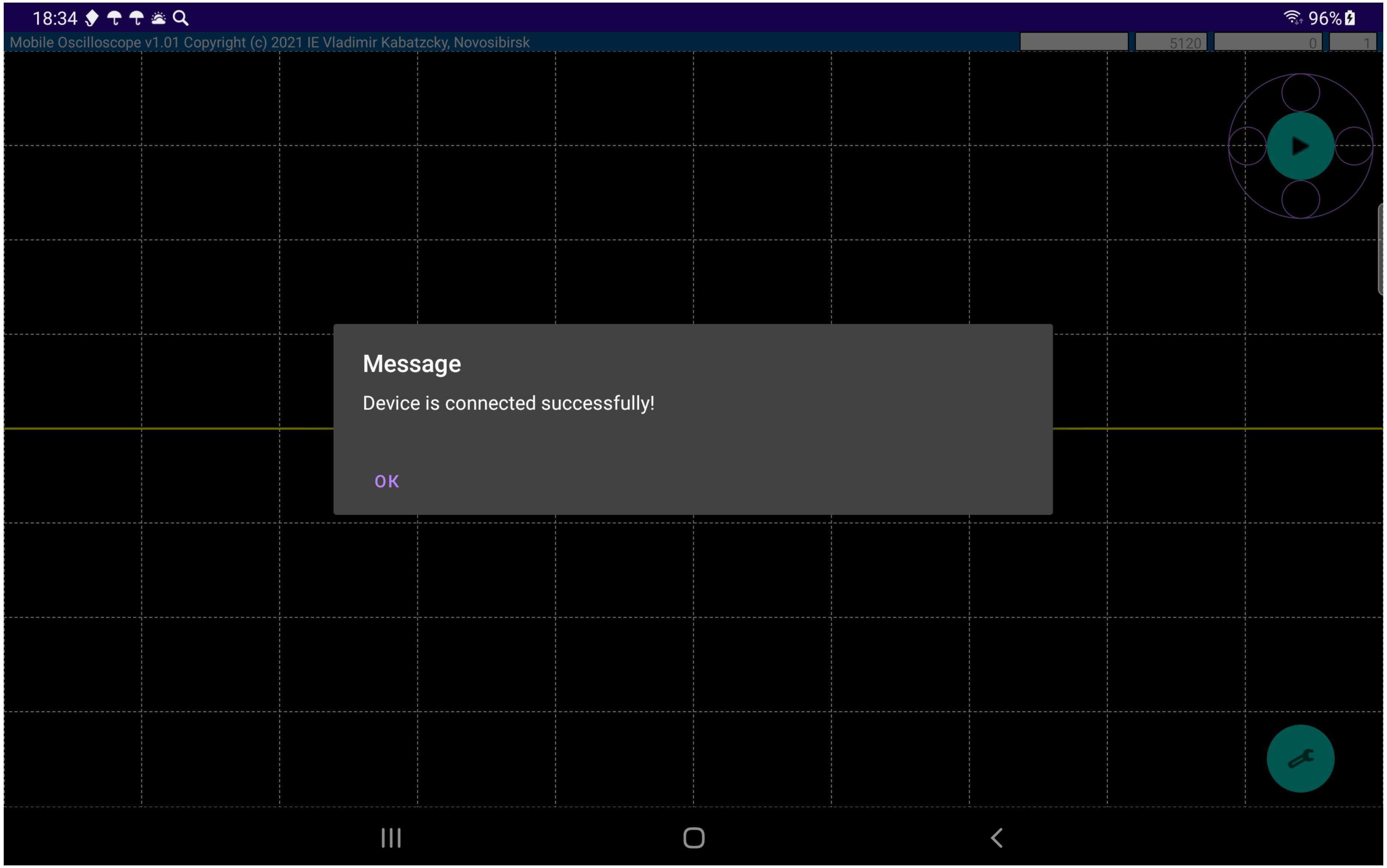
1. Periodic display of accumulated signal samples in the oscilloscope window
2. Scaling along the horizontal and vertical axes
3. The ability to record signal samples in memory and view them in the oscilloscope window
4. Control of the test signal generator
5. Synchronization on the leading or trailing edges of the observed signal
6. Measurement of the amplitude characteristics and frequency of the observed signal

7. Automatic playback (demonstration) of the recorded signal in the form of a smoothly moving graph from right to left
8. Main control is implemented in the form of a single multifunctional key

Features of the single multifunctional key:

1. Turning the oscilloscope on / off
2. Start / stop signal recording
3. Connecting to the observed signal source
4. Vertical and horizontal scaling of the observed and recorded signals
5. Scrolling through the recorded signal
6. Start / stop the automatic playback of the recorded signal

Figure 1. A message about the successful connection to the observed signal source (converter)


18:34
96%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
5120
0
1

Message

Device is connected successfully!

OK

Figure 2. Main window (at the top (right) – multifunctional key, at the bottom (right) – settings and control key). Window width: 2560 px

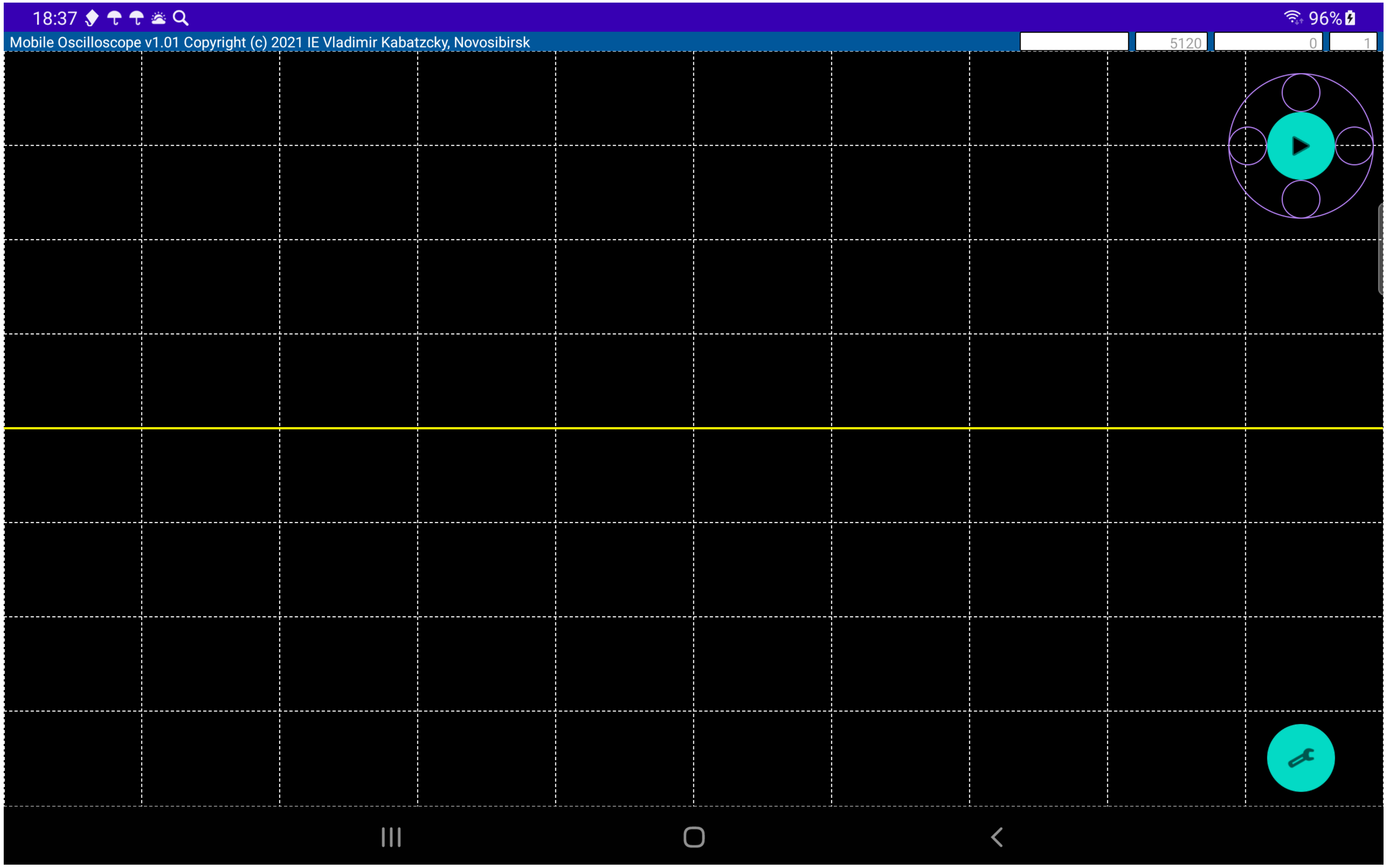
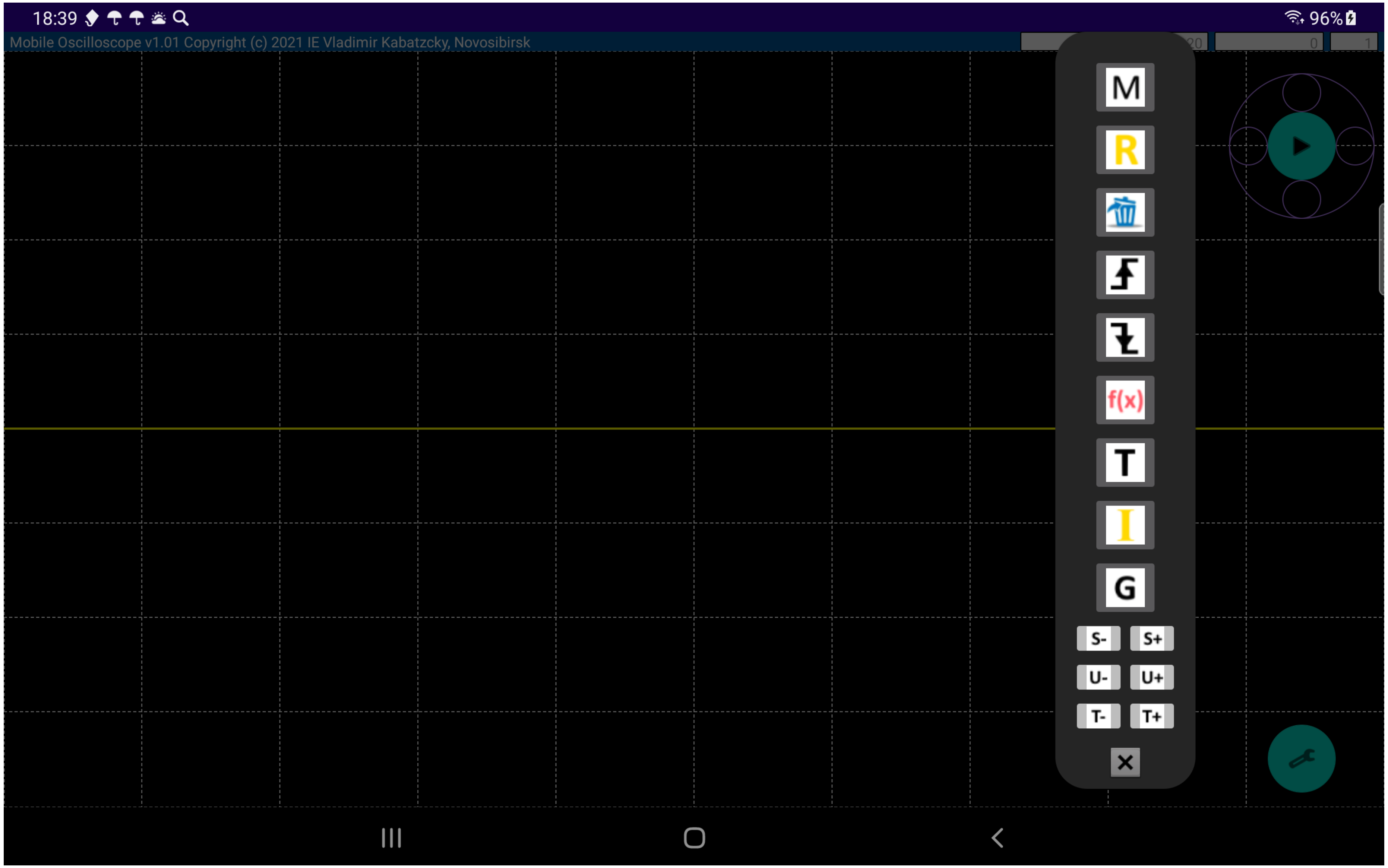
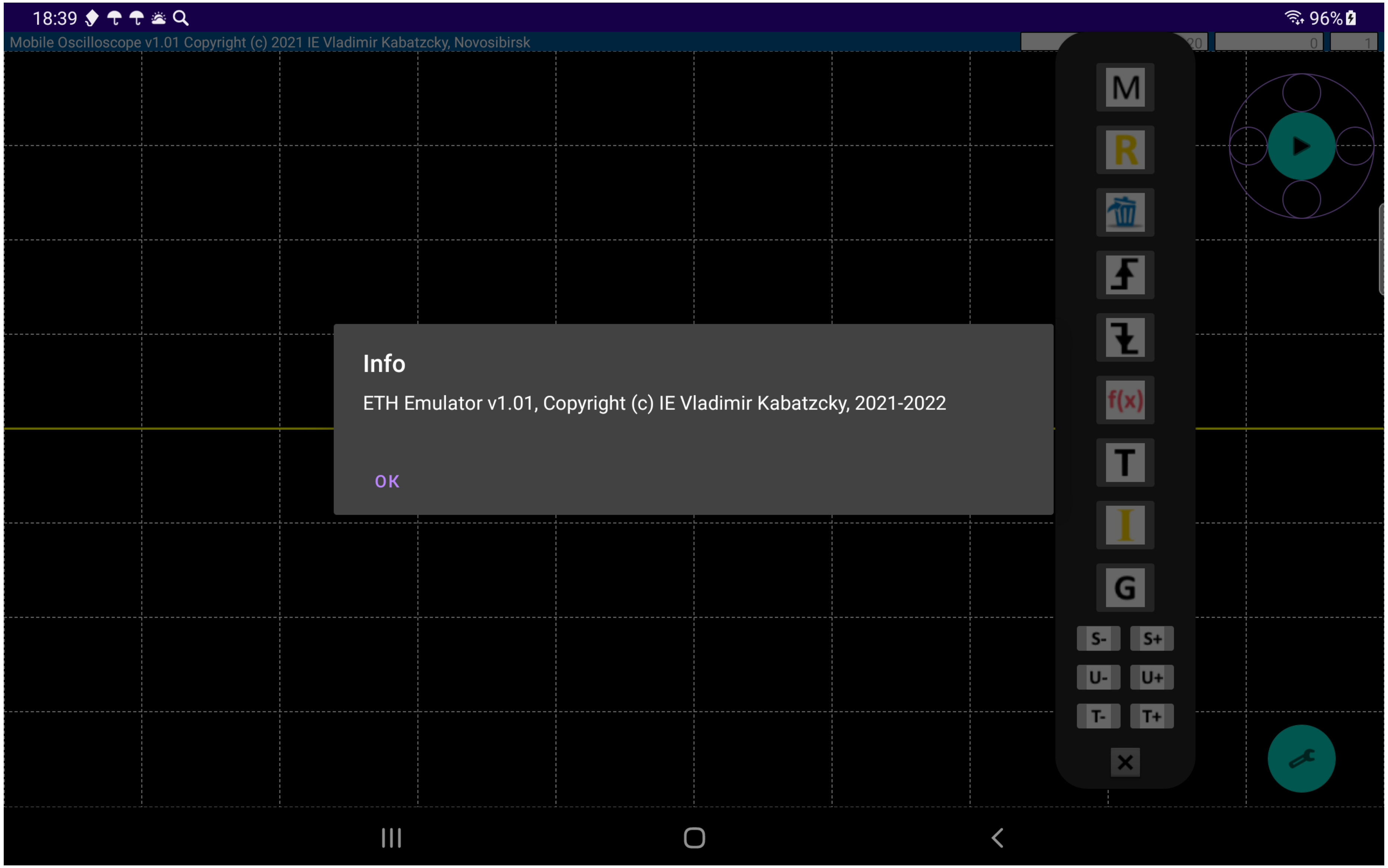

18:37
96%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
5120
0
1

Figure 3. Settings and control window



Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk

Figure 4. Information about the observed signal source


18:39
96%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
M
R
f(x)
T
I
G
S-
S+
U-
U+
T-
T+

Info

ETH Emulator v1.01, Copyright (c) IE Vladimir Kabatzcky, 2021-2022

OK

Figure 5. An observed signal - sinus wave

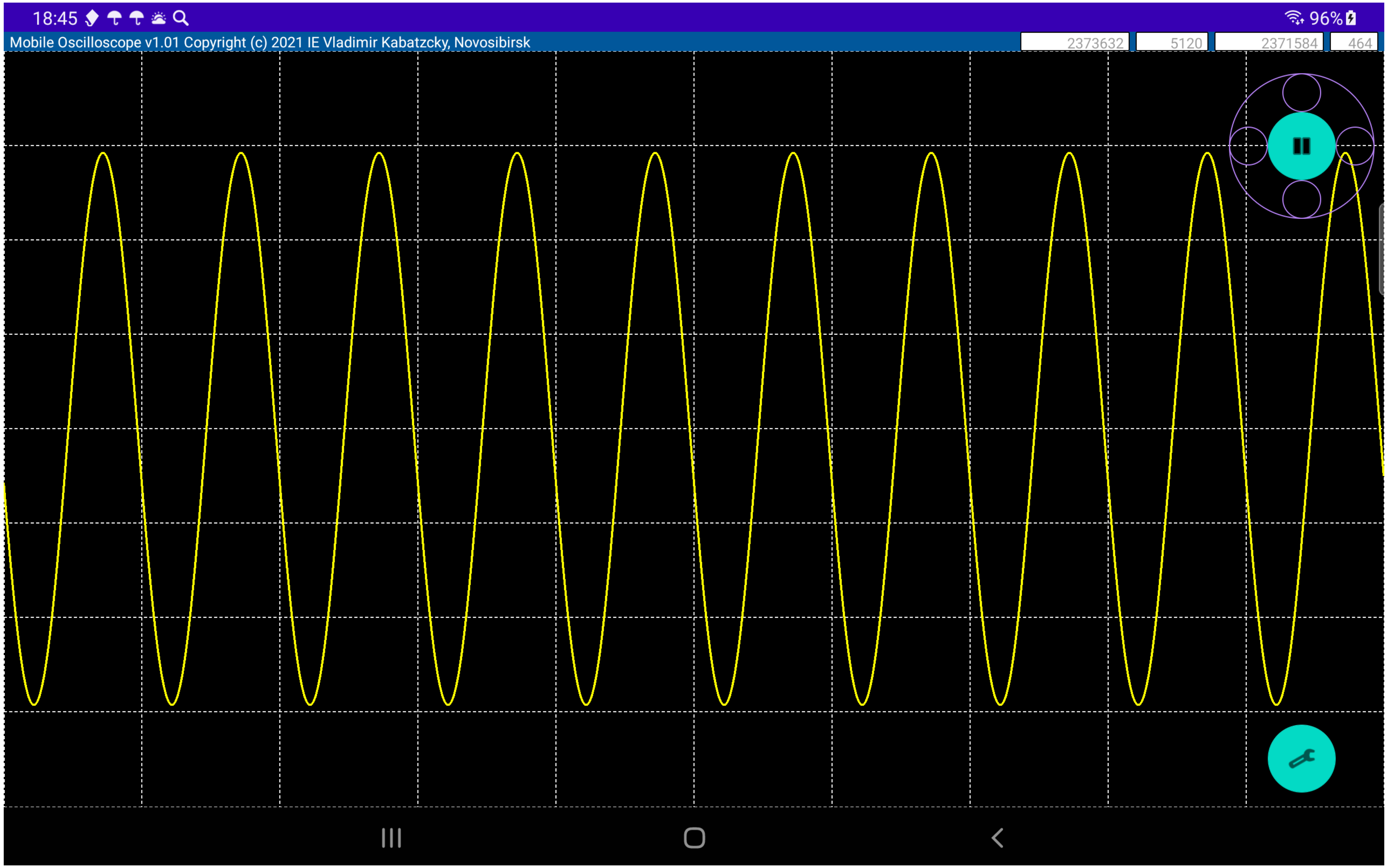

18:45
96%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
2373632
5120
2371584
464

Figure 6. An observed signal - triangle wave

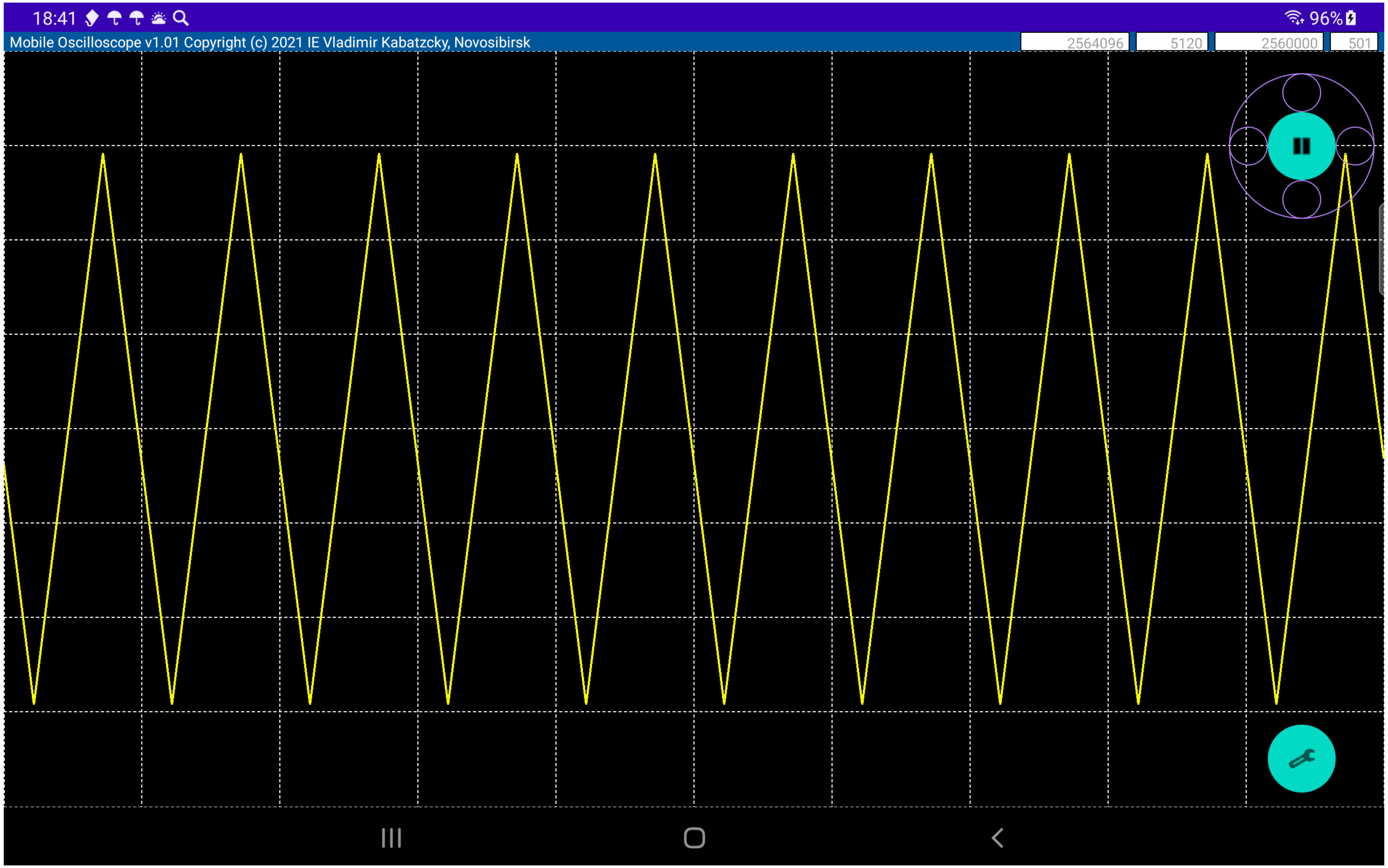

18:41
96%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
2564096
5120
2560000
501

Figure 7. An observed signal - meander wave

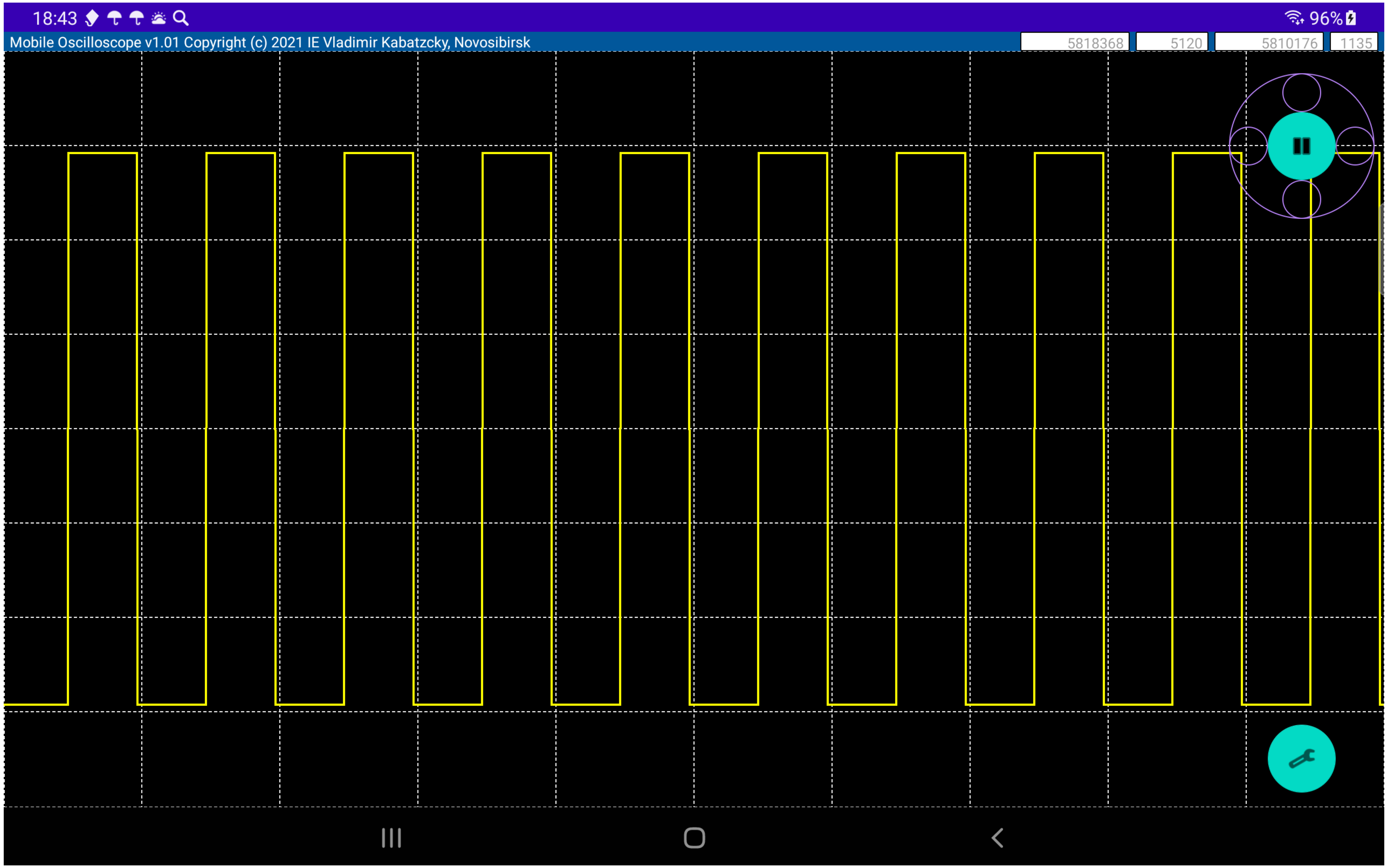
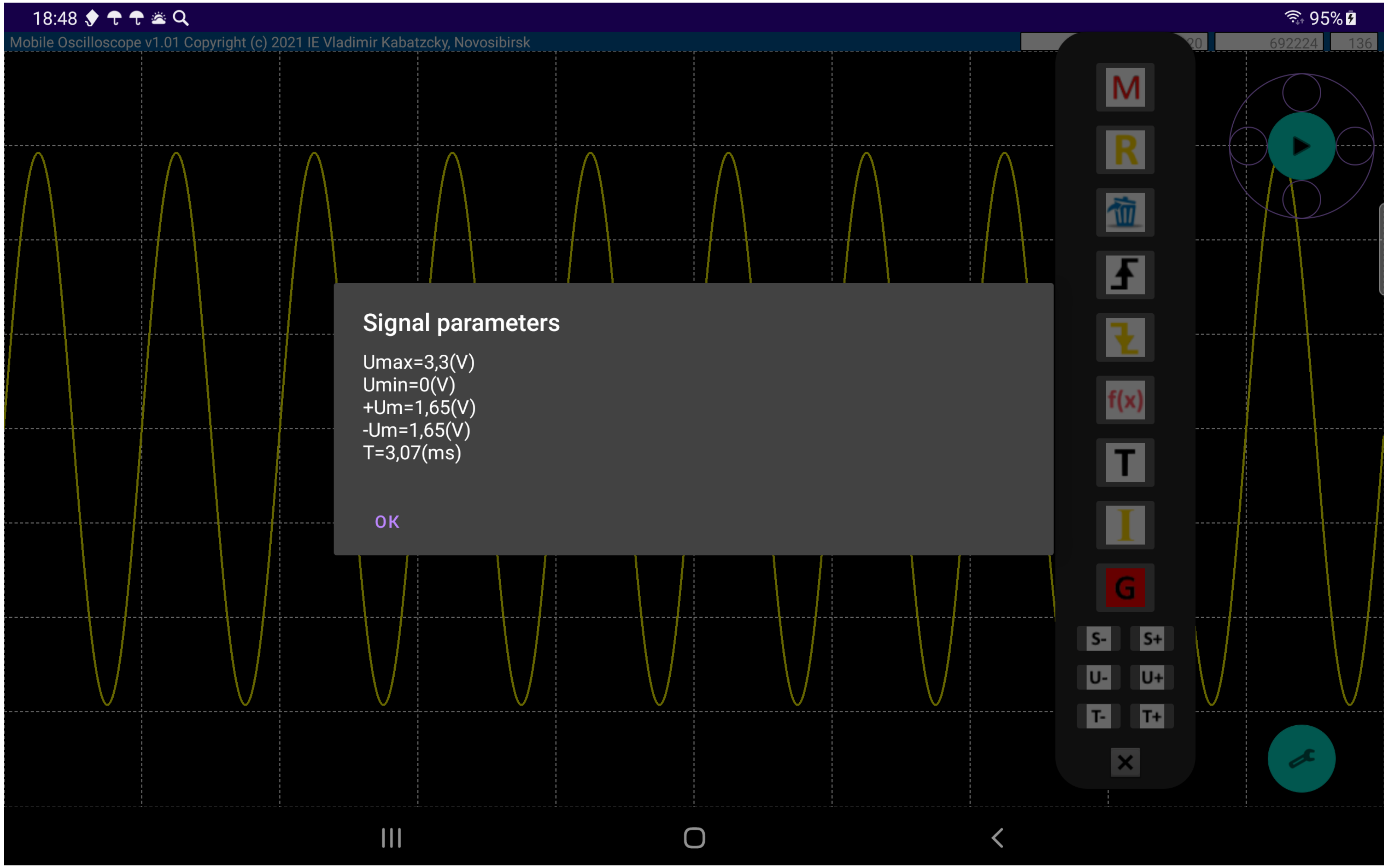

18:43
96%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
5818368
5120
5810176
1135

Figure 8. Measured parameters of the observed signal


18:48
95%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
692224
136
M
R
f(x)
T
I
G
S-
S+
U-
U+
T-
T+
OK

Signal parameters

Umax=3,3(V)
Umin=0(V)
+Um=1,65(V)
-Um=1,65(V)
T=3,07(ms)

Figure 9. Viewing the recorded signal - the moment when the amplitude of the observed signal changes (the scale has been changed - 5120 samples/window)

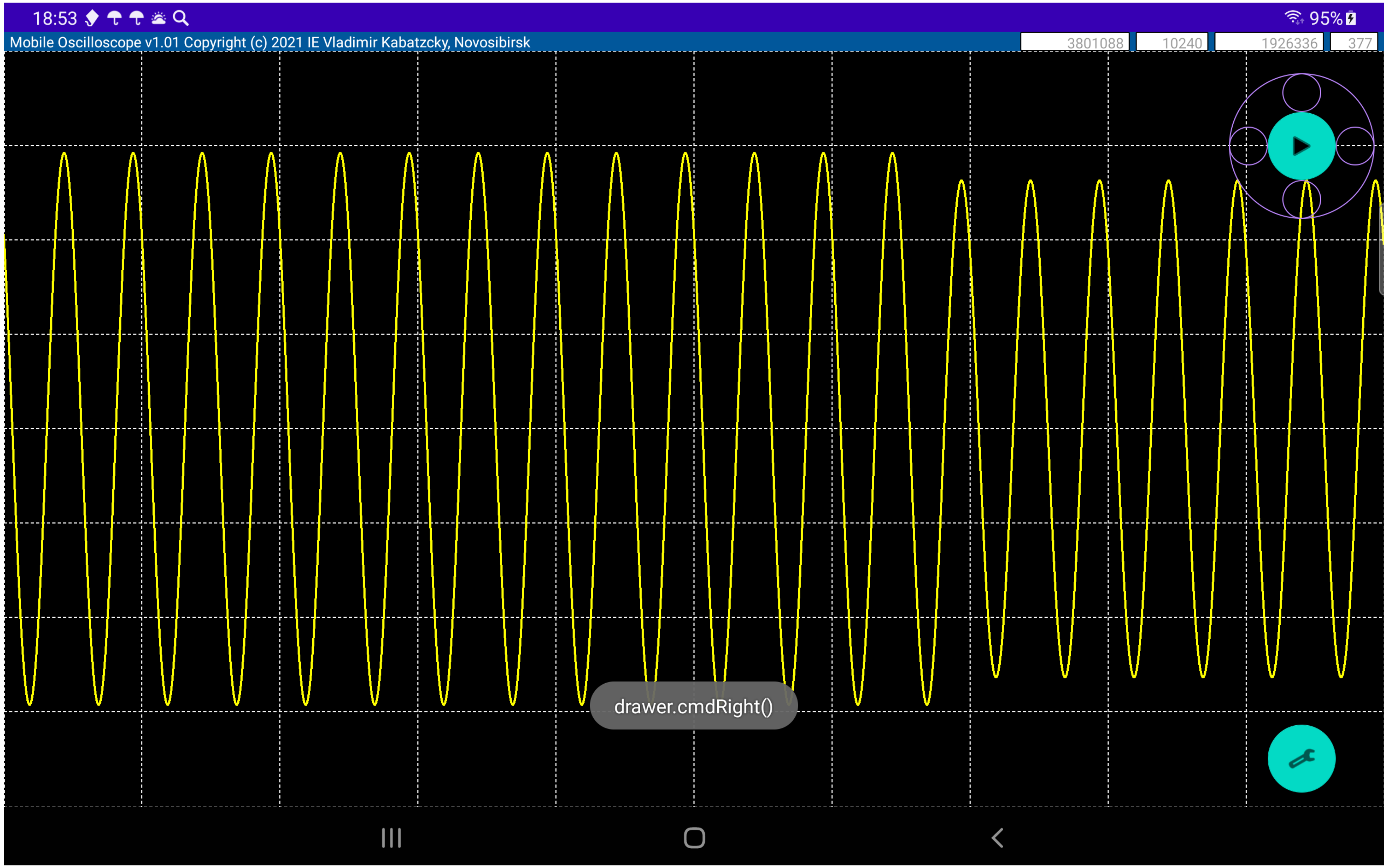

18:53
95%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
3801088
10240
1926336
377
drawer.cmdRight()

Figure 10. Viewing the recorded signal - the moment when the form of the observed signal changes (the scale has been changed - 5120 samples/window)

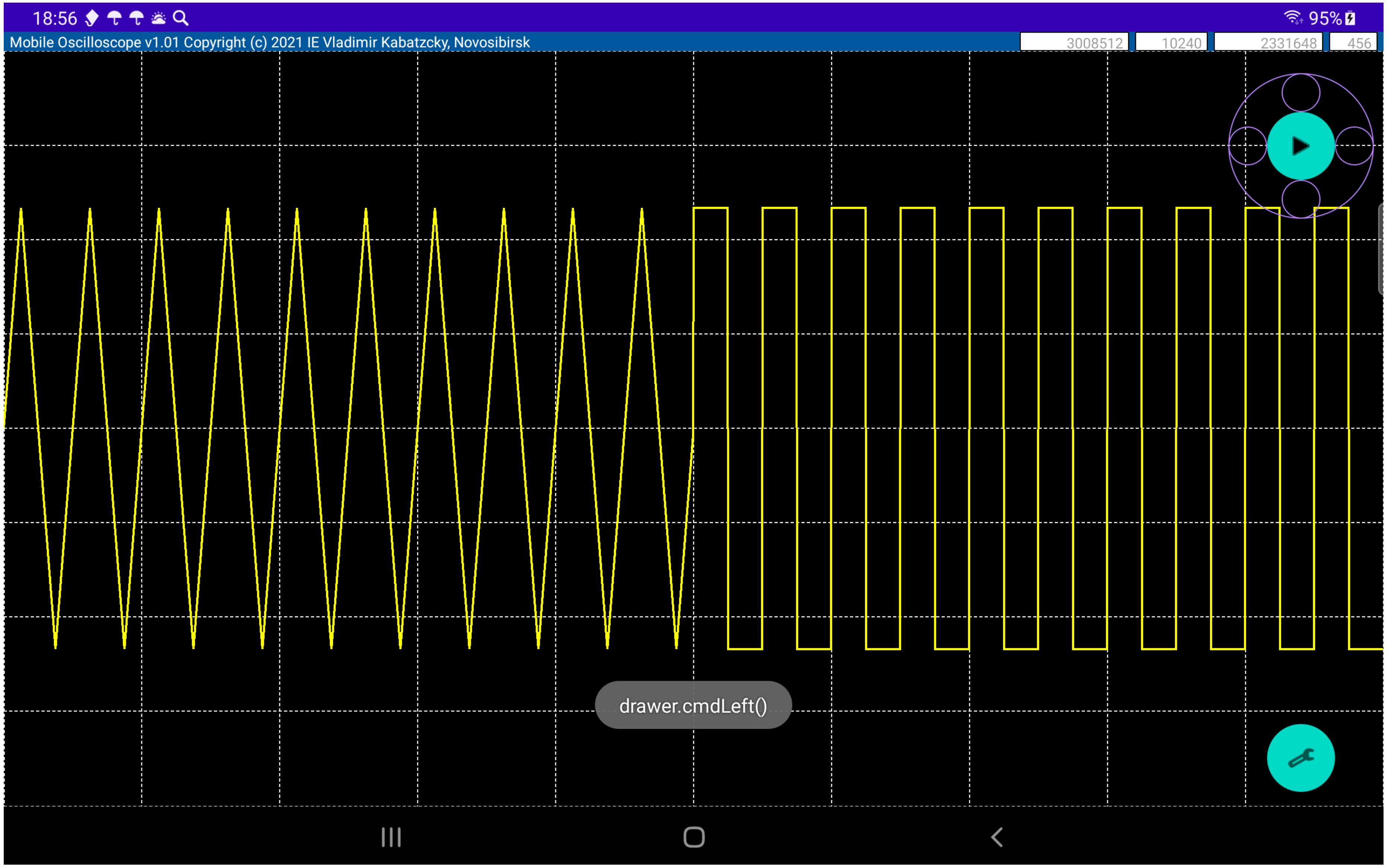

18:56
95%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
3008512
10240
2331648
456
drawer.cmdLeft()

Figure 11. Viewing the recorded signal - the moment where the signal ended (the scale has been changed - 5120 samples/window)

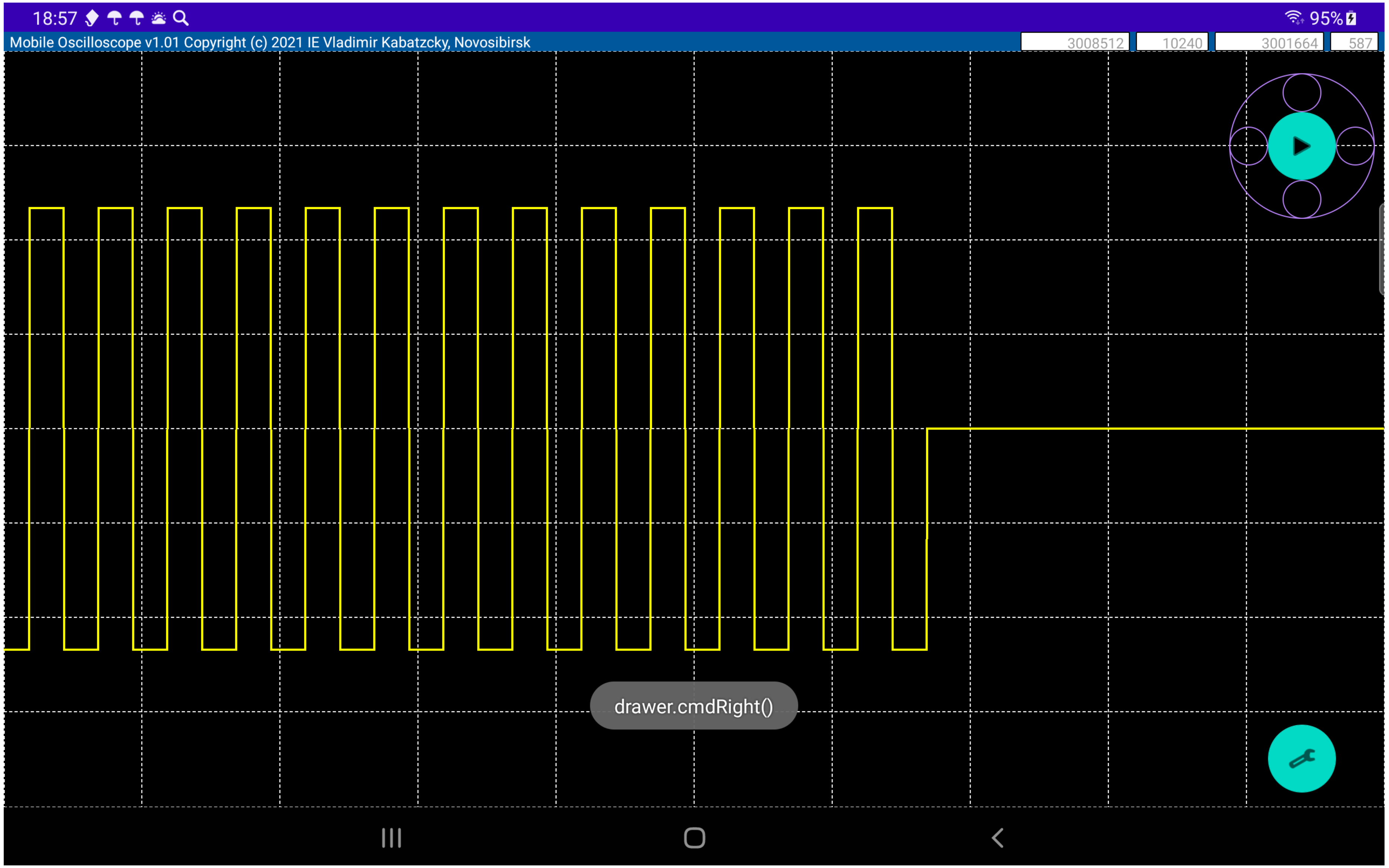

18:57
95%
Mobile Oscilloscope v1.01 Copyright (c) 2021 IE Vladimir Kabatzcky, Novosibirsk
3008512
10240
3001664
587
drawer.cmdRight()

Figure 12. Main activity – code example

```
import android.content.pm.ActivityInfo;
import android.os.Bundle;
import androidx.appcompat.app.ActionBar;
import androidx.appcompat.app.AppCompatActivity;
...

/**
 * @author Copyright (c) 2021 IE Vladimir Kabatzcky
 */
public class MainActivity extends AppCompatActivity {

    private ControlsComponent controller;
    private MobileDevice device;
    private DrawingComponent drawer;
    private MicrophoneData data;
    private Sync sync;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setRequestedOrientation(ActivityInfo.SCREEN_ORIENTATION_REVERSE_LANDSCAPE);
        ActionBar actionBar = getSupportActionBar();
        if (actionBar != null) {
            actionBar.hide();
        }
        setContentView(R.layout.activity_main);

        if (controller == null) {
            controller = new ControlsComponent(MainActivity.this);
            device = new MobileDevice(controller.getResponseHandler());
            drawer = findViewById(R.id.drawer);
            data = MicrophoneData.getInstance();
            sync = Sync.getInstance();

            device.init();

            data.init();
            data.initData();

            controller.setIDevice(device);
            controller.setIData(data);

            sync.setIData(data);

            drawer.setIDevice(device);
            drawer.setIShower(controller);
            drawer.setIPlayer(controller);
            drawer.setIData(data);
            drawer.setISync(sync);
            drawer.init();

            controller.setIDrawer(drawer);

            device.setIDrawer(drawer);
            device.setIShower(controller);
            device.setIData(data);
        }
    }
    @Override
    protected void onResume() {
        super.onResume();
        if (device != null) {
            device.cmdPauseOff();
        }
    }

    @Override
    protected void onPause() {
        super.onPause();
        if (device != null) {
            device.cmdPauseOn();
        }
    }

    @Override
    protected void onDestroy() {
        if (device != null) {
            device.deinit();
        }
        super.onDestroy();
    }
}
```

Thanks for attention!