Heisenbug

Тестируем мобильный телефон по-другому. Полное погружение в HW стенды

Докладчик: Давыдова Надежда

kaspersky

Кто рассказывает?

Надежда Давыдова
Nadezhda.Davydova@kaspersky.com

kaspersky

О чём пойдёт речь

- Зачем нам Continuous Integration (CI)
- Новая ОС и области её применения
- Тестирование в эмуляторе и почему этого недостаточно
- Тестирование с применением оборудования

kaspersky

Зачем нам CI

- Автоматизированный регресс
- Повторяемость
- Стабильность тестов
- Расширение тестового покрытия
- Экономия времени на документировании

KasperskyOS

- Собственная разработка
- Микроядро
- Интеграция с модулем безопасности

kaspersky

Области применения

Kaspersky IoT Infrastructure Security

Комплексная защита инфраструктуры интернета вещей

Kaspersky Secure Remote Workspace

Функциональность и управляемость инфраструктуры тонких клиентов

Kaspersky Automotive Adaptive Platform

Построение надежных IT-систем для умного автотранспорта

kaspersky

Тестирование мобильного телефона на KasperskyOS

kaspersky

Тестирование в эмуляторе

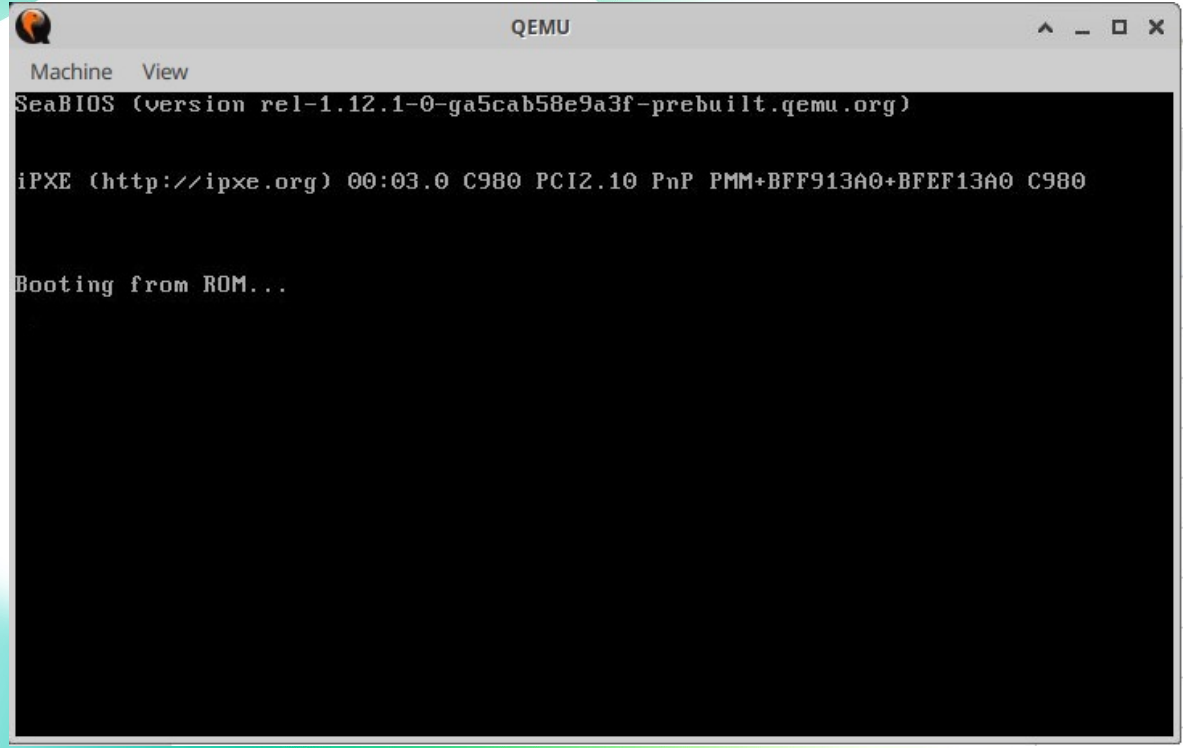
- Что тестируют обычно?
- Почему этого недостаточно?

Что тестируют обычно?

- Работу приложений и сервисов в ОС
- Взаимодействие компонентов

Эмулятор

Эмулятор архитектуры с открытым исходным кодом



kaspersky

Преимущества тестирования в эмуляторе

- Легко интегрировать в CI
- Высокая повторяемость тестов
- Дешевле тестирования на устройстве
- Подходит для различных устройств
- Часть тестов реализована в эмуляторе

Continuous Integration

- Bash скрипты
- Библиотека QMP
- pytest, Python

Пример запуска

```
qemu-system-x86_64 \
-m 1024 \
-cpu core2duo \
-nographic \
-netdev tap,id=net0,ifname=tap0,\
            script=no,downscript=no
-kernel ${PATH}/kos-qemu-image
```

Библиотека QMP

```
- qmp tcp:localhost:4444,server,nowait
```

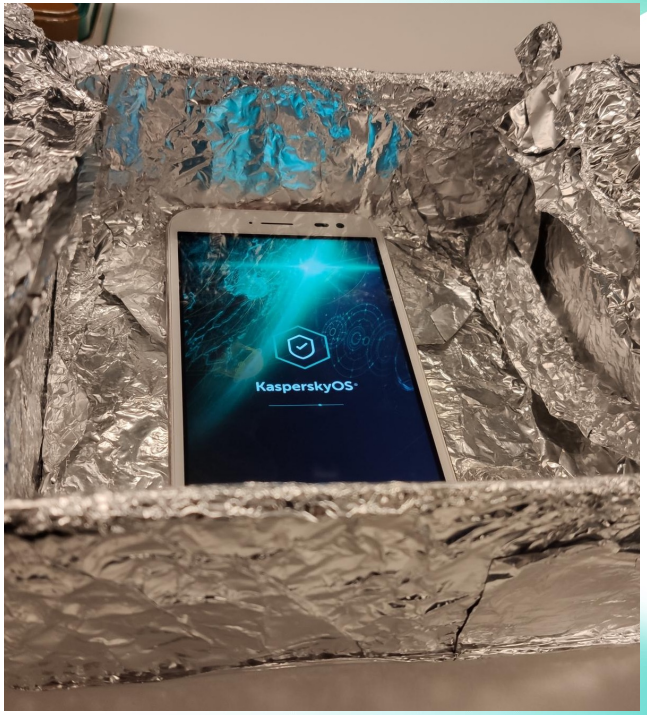
Почему этого недостаточно?

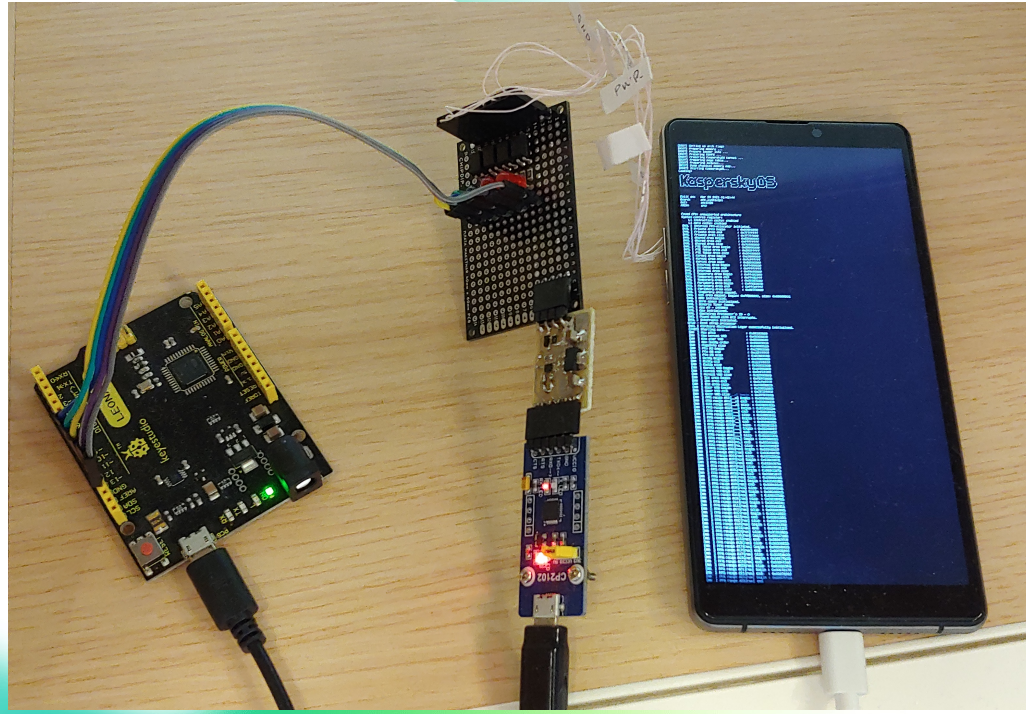
- Трудоёмко писать заглушки реальных устройств
- Не позволяет обнаружить ошибки в работе драйверов
- Телефон может работать по-другому
- Мы тестируем ОС

Что нужно протестировать

- Регистрация в сотовой сети
- Звонки и сообщения
- Передача данных
- Энергопотребление
- Работа периферийных устройств

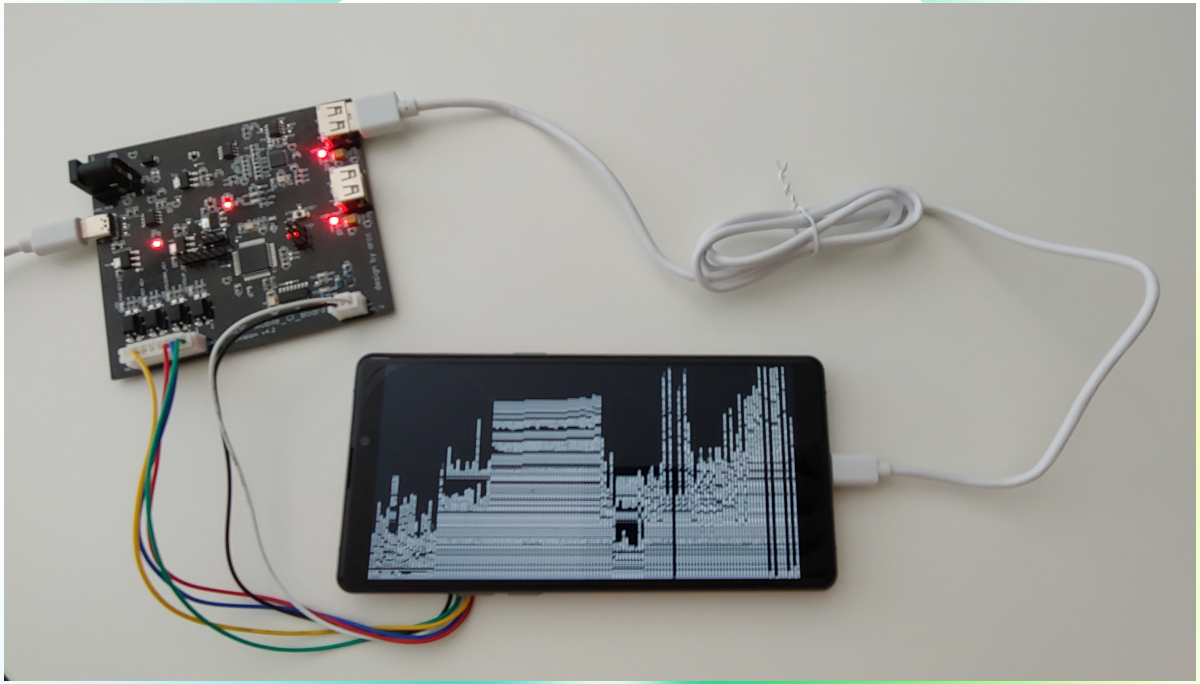
Первые опыты





kaspersky

Прототипирование



kaspersky


HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

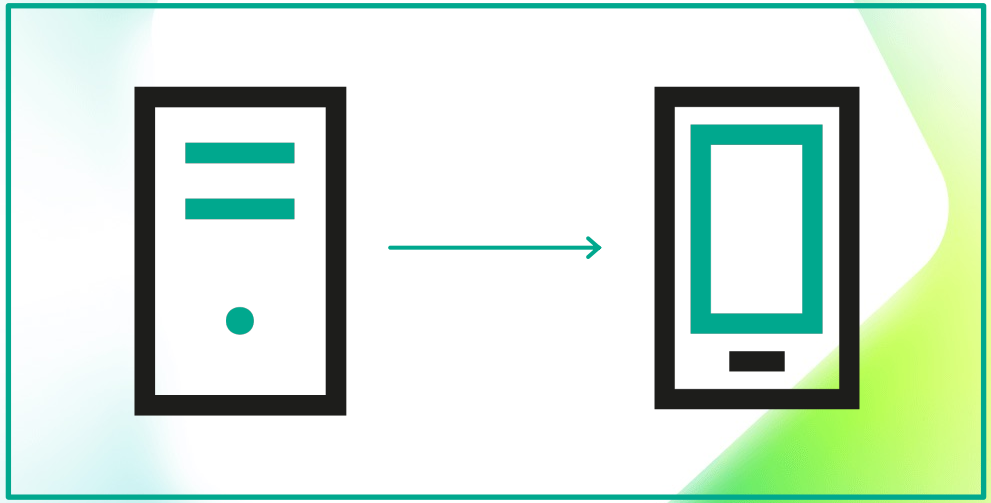
kaspersky


HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

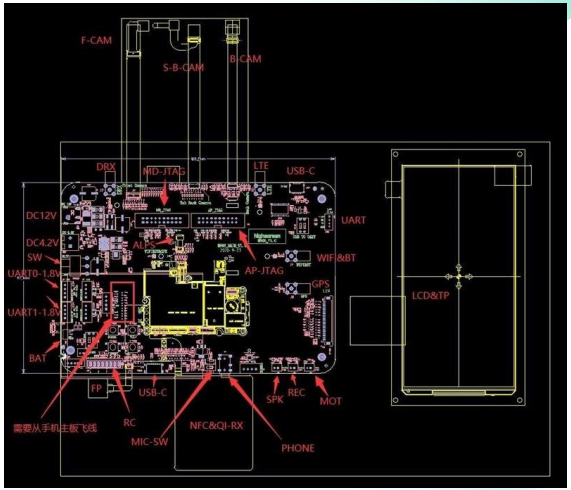
kaspersky

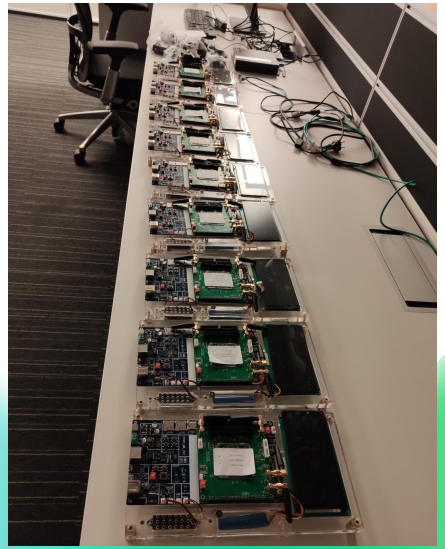
Как доставить сигнал?

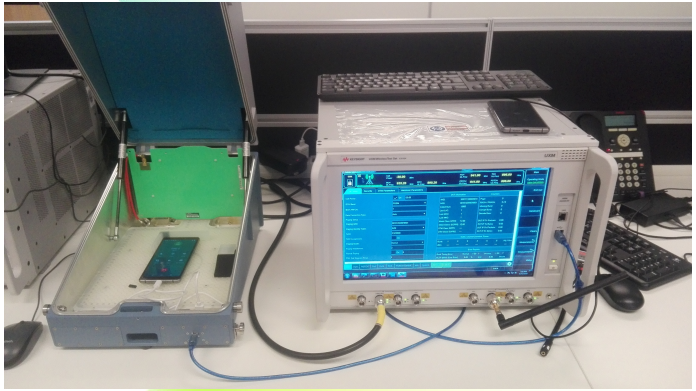
Как доставить сигнал?



Тестирование сотовой сети

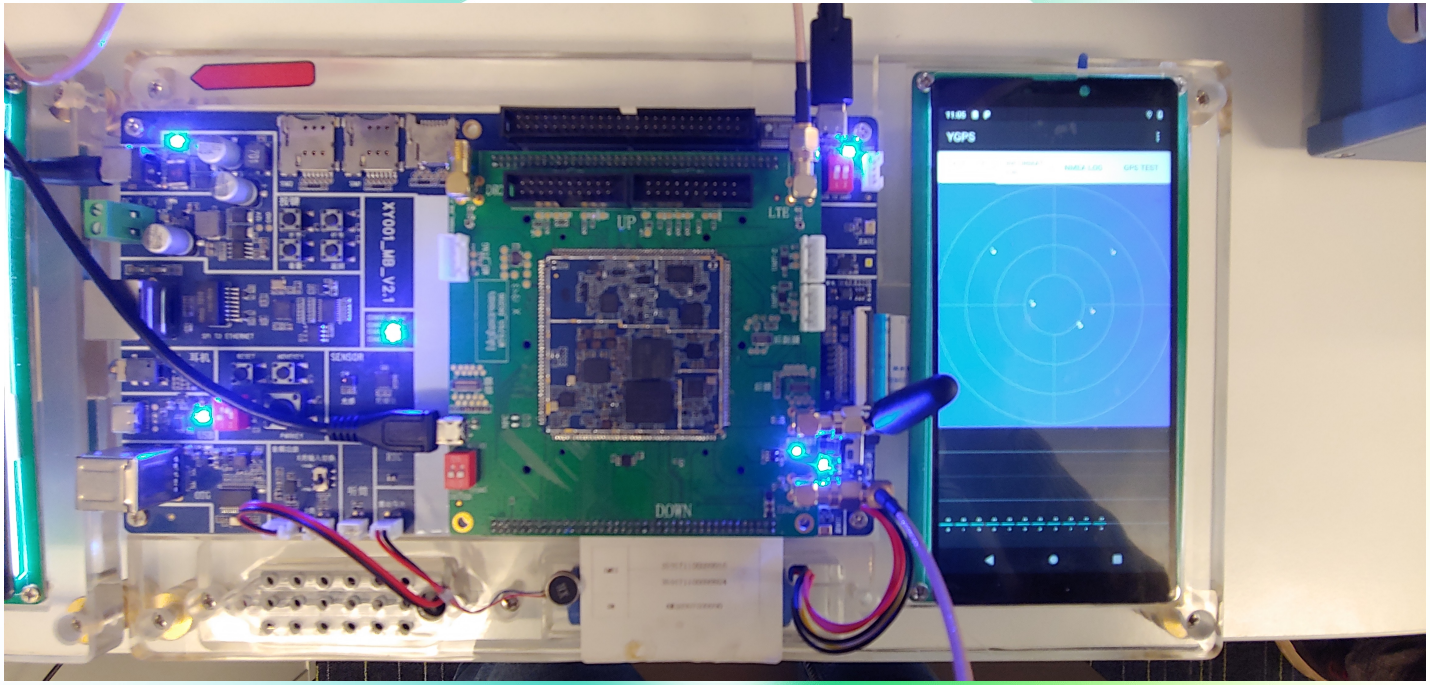




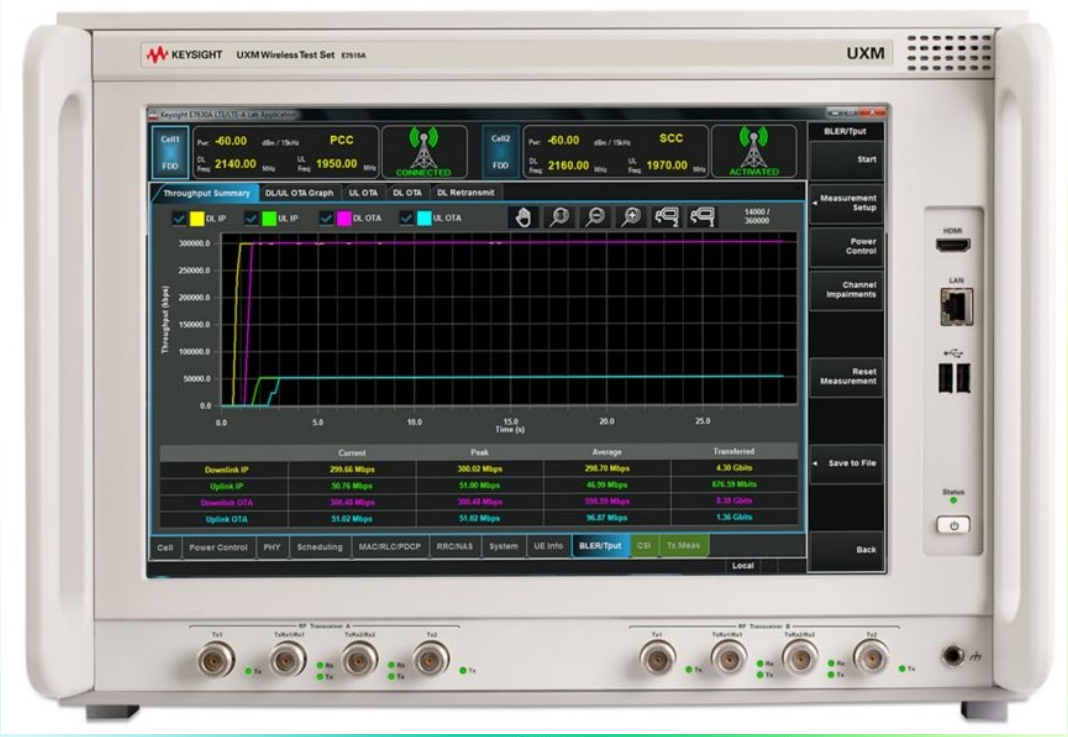


kaspersky

Телефон изнутри



Эмулятор базовой станции



kaspersky

Преимущества

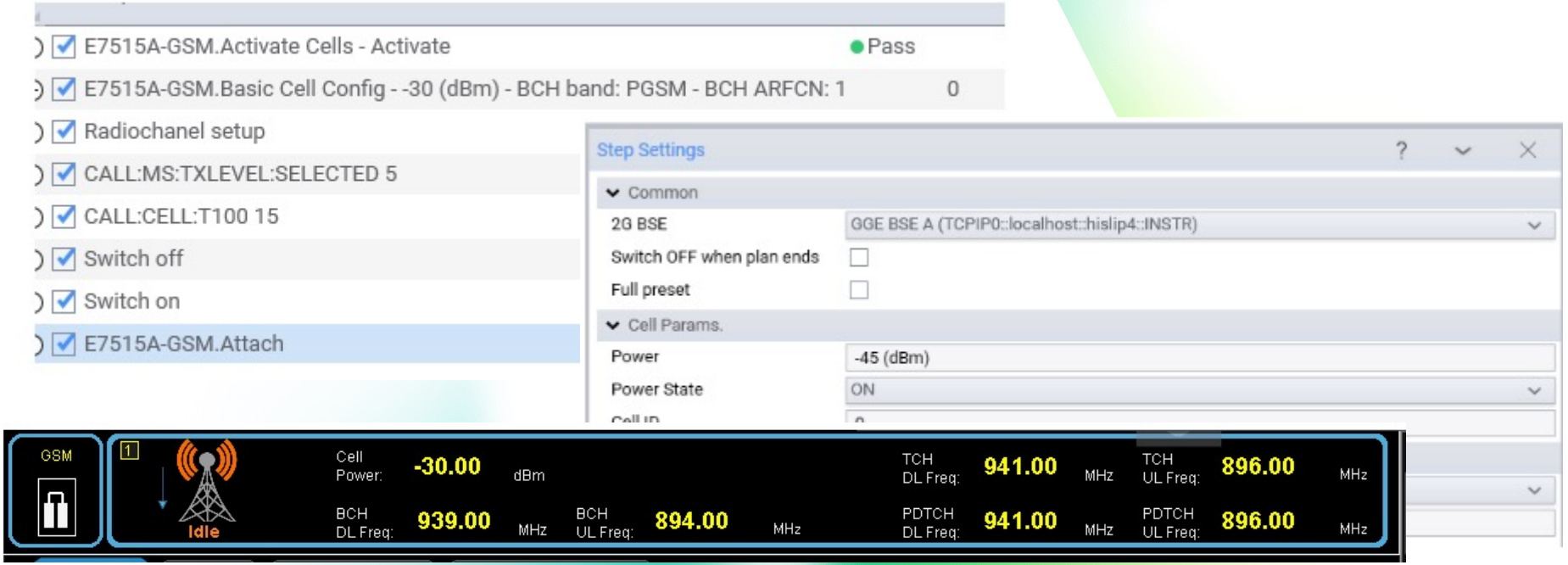
- Комплексное тестирование на телефоне
- Исключение влияния внешней среды
- Детальное исследования определённых режимов работы
- Повторяемость тестов

Интеграция в CI

- Минимизация ручного труда
- Автоматизация управления прибором
- Интеграция в существующую систему

Схема стенда

TAP (Test Automation Platform)



kaspersky

Что следует улучшить?

- Программное управление телефоном
- Запуск последовательности тестов

Android Debug Bridge (adb)

- Основные команды для тестов
- Унификация для нескольких телефонов
- Логирование

ADB – основные команды

```
adb shell settings put global airplane_mode_on 1
adb shell am broadcast -a
android.intent.action.AIRPLANE_MODE --ez state true
adb shell input keyevent 6
adb logcat -b radio
adb shell svc data enable
adb shell am start -a android.intent.action.CALL -d
tel:*102%23
```

Интеграция в CI

Плюсы такого подхода

+ Удобный интерфейс
+ Легко отлаживать
+ Можно запускать из командной строки
+ Конфигурирование прибора

Интеграция в CI

Недостатки

- Формируется очередь
- Долго выполняется тест
- Много ручного труда
- Тесты должны храниться непосредственно на приборе
- Нет управления из Python

OpenTAP – python plugin

$TAP_PATH/Packages/Python/

https://github.com/utepnetlab/opentap

Интеграция в CI

Плюсы

+ Программное управление телефоном
+ Конфигурирование прибора
+ Запуск из Python

Интеграция в CI

Минусы

- Управление через прибор
- Ограничения конфигурирования оборудования
- Не расширяемая схема


HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

kaspersky

HiSLIP + VISA

VISA – Virtual Instrument Software Architecture

HiSLIP – High-Speed LAN Instrument Protocol

pyvisa

```
import pyvisa
def __init__(self, address: str, visa_lib:
str):
    self.rm =
    pyvisa.ResourceManager(visa_lib)
    self.address = address
    self.ControlPanel: ControlPanel = None
    self.TransceiverA: Transceiver = None
    self.TransceiverB: Transceiver = None
```

Подключение к панели управления

```
def control_panel_start(self):

    self.ControlPanel = ControlPanel(self.rm,
    'TCPIP0::' + self.address +
    '::hislip0::INSTR’)
    logging.info(self.ControlPanel, "Control
    Panel hasn't been connected")
```

Настройка трансивера

```
def configuration_set_3g(self, power_cell: int = -50,
                                    cable_correction: int = 0):
    self.TransceiverA = Transceiver3G(rm=self.rm,
    port='TCPIP0::' + self.address + '::hislip4::INSTR’)
    logging.info(self.TransceiverA, "Transceiver hasn't
    been connected“)
    self.TransceiverA.configuration_set(power_cell=power_c
    ell, cable_correction=cable_correction)
```

kaspersky

Что-то знакомое

```
div > div:nth-of-type(2) div > div:nth-child(2)

#main-panel > div.slideshow-info-container >
div.notranslate.transcript.add-padding-right.j-transcript
> ol > li:nth-child(1)
```

kaspersky

SCPI - Standard Commands for Programmable Instruments

```
SOURce:POWer[:LEVel]:SLOPe[:DATA]

SENSe:FREQuency:CENTer


CALC:LIMit:DATA
2,1,1E9,3E9,0,0,2,1E9,3E9,-3,-3
```

- Условные обозначения
- Формат команд
- Параметризация команд
- Запрос настроек параметров
- Терминаторы команд SCPI

kaspersky

SCPI параметризация команд

```
SENSe:FREQuency:STARt 1 MHz
SENSe:FREQuency:STARt?

DISPlay:ENABle OFF

MMEMory:STORe "state01.sta"
```

Типы параметров SCPI

- Числовые параметры
- Дискретные параметры
- Булевы параметры
- Параметры строк ASCII

kaspersky

Пример использования SCPI команд для отправки SMS

```
def __call_sms_ptp_text_cust(self, text: str):

    ans =
    self.instrument.write('CALL:SMService:PTPoin
    t:TEXT:CUSTom "%s"' % text)
```

kaspersky

Пример использования SCPI команд для осуществления вызовов

```
def __call_orig(self):
    ans =
    self.instrument.write('CALL:ORIGinate’)

def __call_end(self):
    ans = self.instrument.write('CALL:END')
```

kaspersky

Интеграция в CI

Плюсы

+ Управление по сети
+ Распараллеливание работ
+ Подключение к CI

Интеграция в CI

Минусы

- Ручная отладка
- Высокий порог вхождения в тесты

Путь к автоматизации

- Построение стендов
- Ручное тестирование
- Проверка тестов на телефоне под управлением Android
- Автоматизация тестов через TAP
- Переход к OpenTAP Python
- Удалённое управление прибором при помощи SCPI команд, HiSLIP + VISA протоколов


HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

kaspersky


HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

kaspersky

Что тестируем

- Точность определения координат
- Время обнаружения спутников
- Различные конфликтные ситуации

GPS эмулятор

Тестирование работы с геопозицией.

GPS эмулятор

```
def set_location(self, lat: float, lon: float, hgt: float):

        self.process = subprocess.Popen([self.path + 'GPS_emulator’,
                                         '-e', self.path + self.brdc,
                                         '-l', ','.join([str(lat),
                                         str(lon), str(hgt)])],
                                         stdout=subprocess.PIPE)

        output = self.process.stdout.readline().decode("UTF-8")
        info(output)
        while "GPS signal generator is ready" not in output:
                output = self.process.stdout.readline().decode("UTF-8")
                info(output)
```

kaspersky

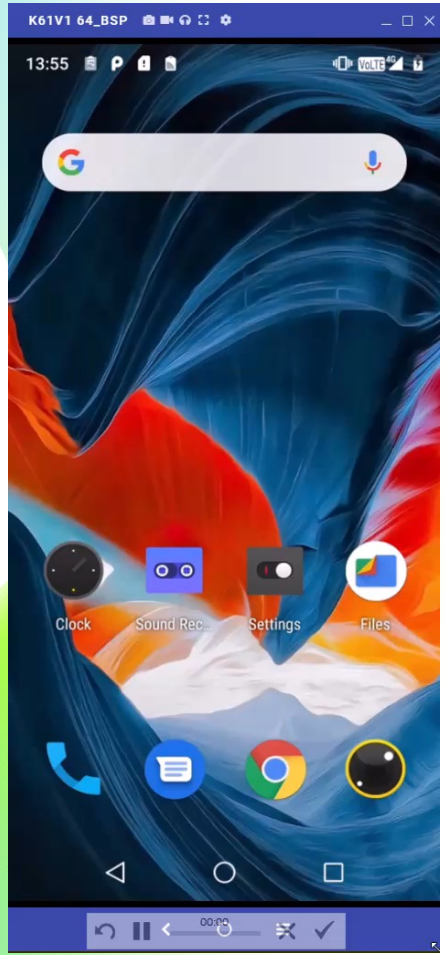
GPS эмулятор

```
Tests passed: 0 of 1 test
DEBUG    root:__init__.py:50 Broadcasting: Intent { act=android.intent.action.AIRPLANE_MODE flg=0x400000 (has extras) }
                             Security exception: Permission Denial: not allowed to send broadcast android.intent.action.AIRPLANE_MODE from pid=13257, uid=2000

                             java.lang.SecurityException: Permission Denial: not allowed to send broadcast android.intent.action.AIRPLANE_MODE from pid=13257, uid=2000
                                at com.android.server.am.ActivityManagerService.broadcastIntentLocked(ActivityManagerService.java:21537)
                                at com.android.server.am.ActivityManagerService.broadcastIntent(ActivityManagerService.java:22174)
                                at com.android.server.am.ActivityManagerShellCommand.runSendBroadcast(ActivityManagerShellCommand.java:698)
                                at com.android.server.am.ActivityManagerShellCommand.onCommand(ActivityManagerShellCommand.java:174)
                                at android.os.ShellCommand.exec(ShellCommand.java:103)
                                at com.android.server.am.ActivityManagerService.onShellCommand(ActivityManagerService.java:16216)
                                at android.os.Binder.shellCommand(Binder.java:634)
                                at android.os.Binder.onTransact(Binder.java:532)
                                at android.app.IActivityManager$Stub.onTransact(IActivityManager.java:3569)
                                at com.android.server.am.ActivityManagerService.onTransact(ActivityManagerService.java:3350)
                                at android.os.Binder.execTransact(Binder.java:731)
DEBUG    root:__init__.py:57
DEBUG    root:__init__.py:59 Broadcasting: Intent { act=android.intent.action.AIRPLANE_MODE flg=0x400000 (has extras) }
                             Security exception: Permission Denial: not allowed to send broadcast android.intent.action.AIRPLANE_MODE from pid=13282, uid=2000

                             java.lang.SecurityException: Permission Denial: not allowed to send broadcast android.intent.action.AIRPLANE_MODE from pid=13282, uid=2000
                                at com.android.server.am.ActivityManagerService.broadcastIntentLocked(ActivityManagerService.java:21537)
                                at com.android.server.am.ActivityManagerService.broadcastIntent(ActivityManagerService.java:22174)
                                at com.android.server.am.ActivityManagerShellCommand.runSendBroadcast(ActivityManagerShellCommand.java:698)
                                at com.android.server.am.ActivityManagerShellCommand.onCommand(ActivityManagerShellCommand.java:174)
                                at android.os.ShellCommand.exec(ShellCommand.java:103)
                                at com.android.server.am.ActivityManagerService.onShellCommand(ActivityManagerService.java:16216)
                                at android.os.Binder.shellCommand(Binder.java:634)
                                at android.os.Binder.onTransact(Binder.java:532)
                                at android.app.IActivityManager$Stub.onTransact(IActivityManager.java:3569)
                                at com.android.server.am.ActivityManagerService.onTransact(ActivityManagerService.java:3350)
                                at android.os.Binder.execTransact(Binder.java:731)
INFO     root:__init__.py:46 Opening and initializing device...
INFO     root:__init__.py:49 deviceName: LimeSDR-USB
```

PyCharm 2020.2.5 available
Update...



kaspersky

Интеграция в CI

Плюсы

+ Автоматизированное управление GPS эмулятором
+ Открытый исходный код

kaspersky

Интеграция в CI

Минусы

- Новый протокол взаимодействия
- Низкая точность геопозиционирования
- Высокие требования к ПК

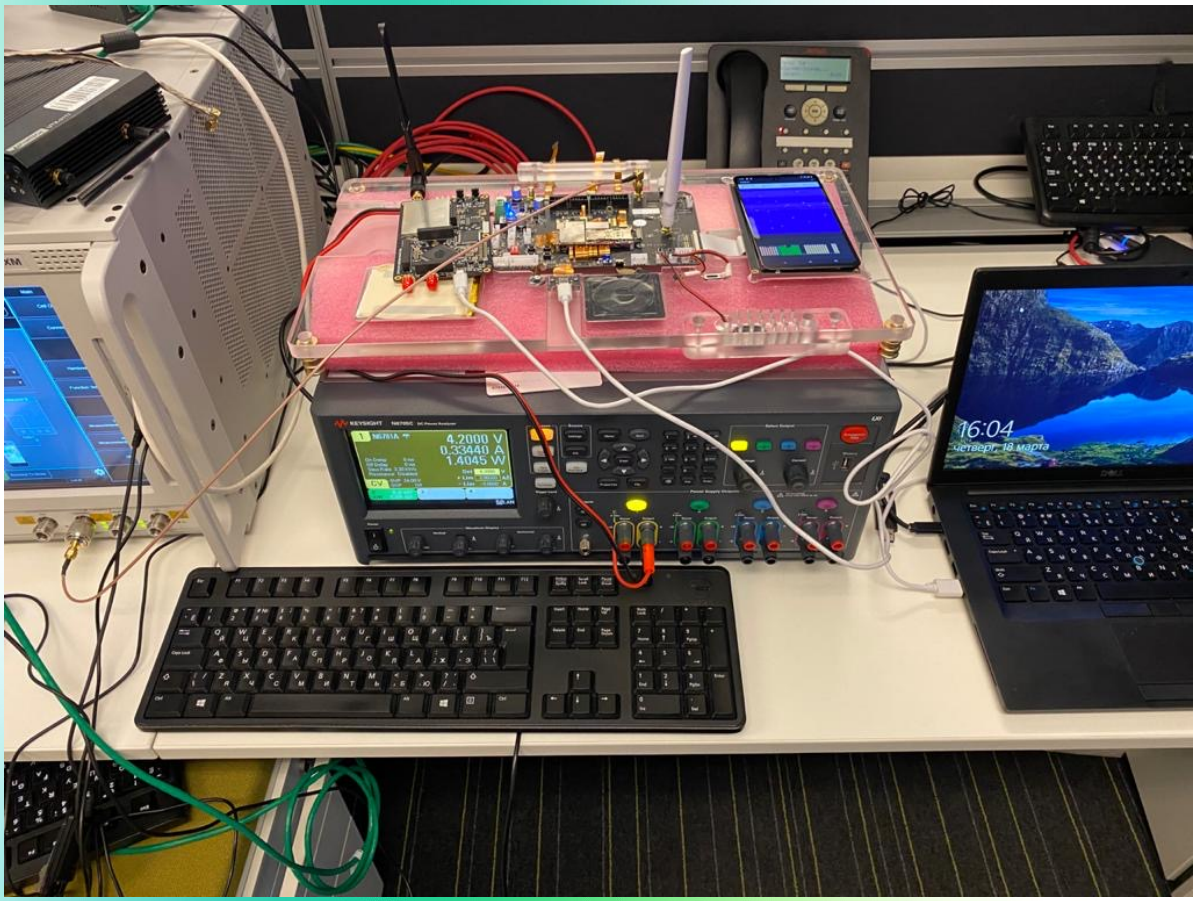

HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

kaspersky

Что тестируем

- Потребление узлов
- Работу при граничных уровнях заряда
- Работа зарядки
- Правильная работа ОС с аккумулятором

Power Monitor



kaspersky

Power Monitor

- Оценка энергопотребления узлов
- Имитация батареи
- Имитация зарядного устройства
- Повторяемость
- Интеграция в CI

Интеграция в CI

Плюсы

+ Автоматизированное управление по сети
+ Высокая повторяемость тестов
+ Независимость от износа телефона
+ Моделирование нестабильностей и сложных сценариев

kaspersky

Интеграция в CI

Минусы

- Требуются отладочные платы
- Высокая стоимость


HW под управлением KasperskyOS
2G, 3G, LTE
LOGS
Bluetooth/Wi-Fi
GPS/GLONASS
Power Supply
CI

kaspersky

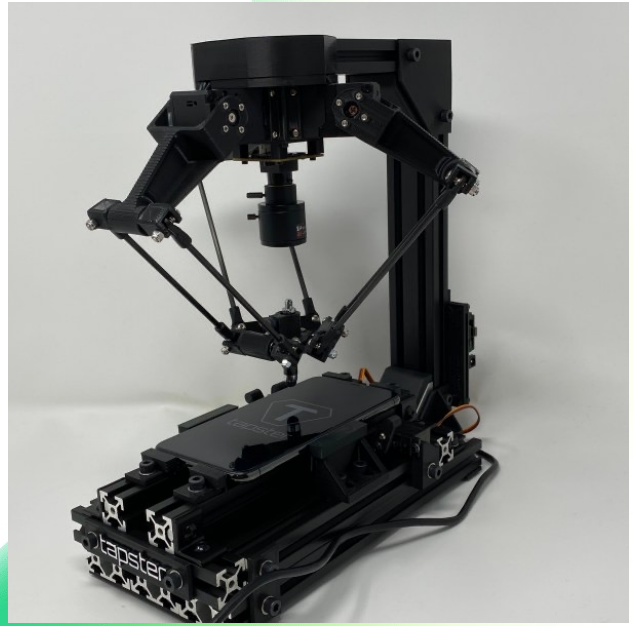
Bluetooth, WiFi emulator

- Контроль качества работы драйверов
- Исключение влияния окружающей среды
- Проверка граничных значений
- Моделирование нестандартных ситуаций

Bluetooth и Wi-Fi

Сквозные тесты

- Все пользуются телефоном по-разному
- Научим конвейер повторять работу с экраном



Выводы

- Начинать с эмуляторов
- Провести глубокий анализ тестовых сценариев
- Применять общий подход к оборудованию (VISA + HiSLIP)
- Строить CI одинаково для различных проектов
- Следить за разумной автоматизацией

Выводы

- Эмуляторы используются для CI
- Генераторы управляются по сети
- Схема стендов расширяемая
- Высокая повторяемость тестов
- Сложные сценарии тестирования
- Проверки с высокой точностью
- Регресс автоматизирован

Ссылки

https://os.kaspersky.ru/
https://www.qemu.org/
https://wiki.qemu.org/Documentation/QMP
https://pypi.org/project/qmp/
https://github.com/utepnetlab/opentap
https://en.wikipedia.org/wiki/Standard_Commands_for_Programmable_Instruments
https://habr.com/ru/article/499746/#rec186469108

Q&A

Questions and Answers