

Регистрация присутствия и биометрии пользователя по протоколу Bluetooth в GNU/Linux

Биометрические данные

- Статические:
 - отпечатки пальцев,
 - распознавание лица,
 - ...
- Динамические:
 - частота сердечных сокращений,
 - артериальное давление,
 - электроэнцефалограмма

Что опробованно на сегодня?

- **Энцефалографы:**
 - NeuroSky MindWave,
 - NeuroSky MindSet,
 - Emotiv EPOC/EPOC+,
 - Emotiv Insight
- **Фитнес-трекеры:**
 - Fibit Charge HR,
 - Xiaomi Mi Band 2/3,
 - Amazfit Bip

Фитнес-трекеры



Портативные энцефалографы



Способы получения данных от Bluetooth-устройств

- 1) использование универсального API;
- 2) использование API, который предоставляется вендором устройства;
- 3) применение специальных утилит для получения данных от конкретного устройства

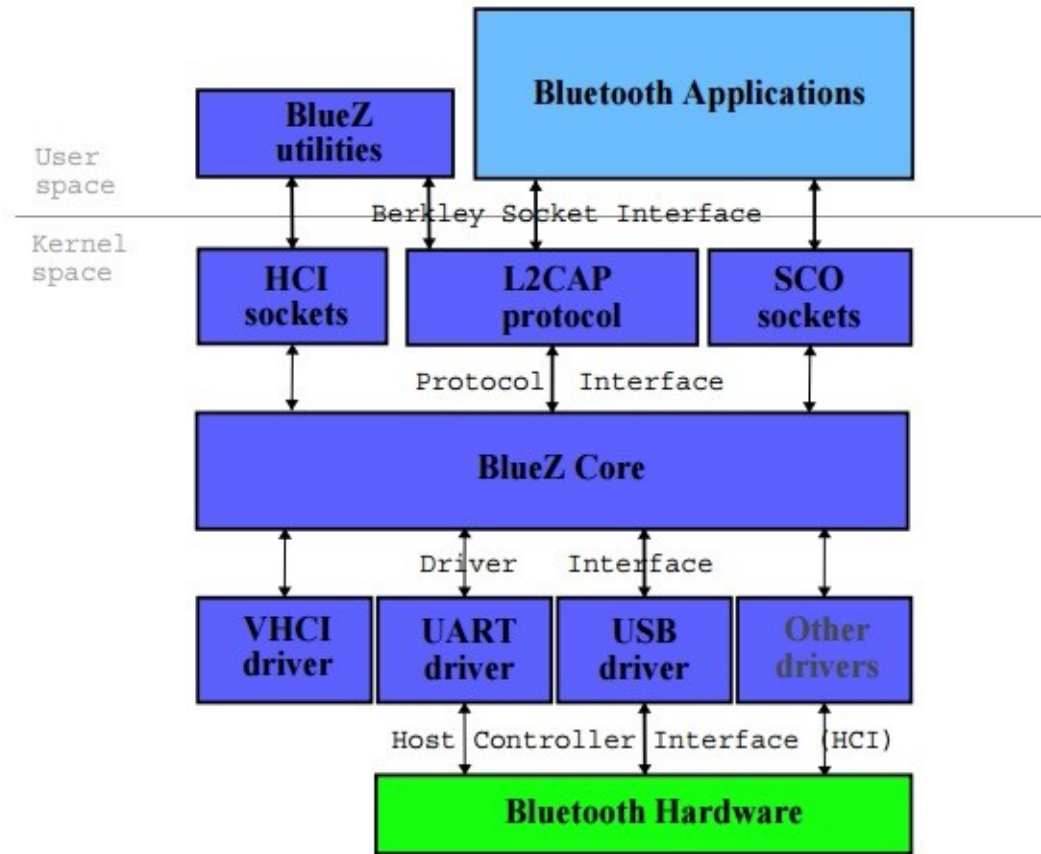
Способы получения данных от Bluetooth-устройств


1) использование универсального API;

2) использование API, который предоставляется вендором устройства;

3) применение специальных утилит для получения данных от конкретного устройства

Реализация в Linux: BlueZ



 - Components provided by BlueZ

Bluetooth и BLE

	Bluetooth	Bluetooth LE
Радиочастота	2.4 ГГц	2.4 ГГц
Расстояние	100 м	>100 м
Пропускная способность	0.7-2.1 Mb/s	0.27 Mb/s
Используемая мощность	1 Вт	0,01 - 0,5 Вт
Максимальный используемый ток	<30 мА	<15 мА

Поиск BLE устройств

```
from bluepy.btle import Scanner, DefaultDelegate

class ScanDelegate(DefaultDelegate):
    def __init__(self):
        DefaultDelegate.__init__(self)

    def handleDiscovery(self, dev, isNewDev, isNewData):
        if isNewDev:
            print "Discovered device", dev.addr
        elif isNewData:
            print "Received new data from", dev.addr

scanner = Scanner().withDelegate(ScanDelegate())
devices = scanner.scan(10.0)

for dev in devices:
    print "Device %s (%s), RSSI=%d dB" % (dev.addr, dev.addrType, dev.rssi)
    for (adtype, desc, value) in dev.getScanData():
        print "  %s = %s" % (desc, value)
```

QtBluetooth

```
void MyClass::startDeviceDiscovery() {
    // Create a discovery agent and connect to its signals
    QBluetoothDeviceDiscoveryAgent *discoveryAgent = new
QBluetoothDeviceDiscoveryAgent(this);
    connect(discoveryAgent, SIGNAL(deviceDiscovered(QBluetoothDeviceInfo)),
this, SLOT(deviceDiscovered(QBluetoothDeviceInfo)));

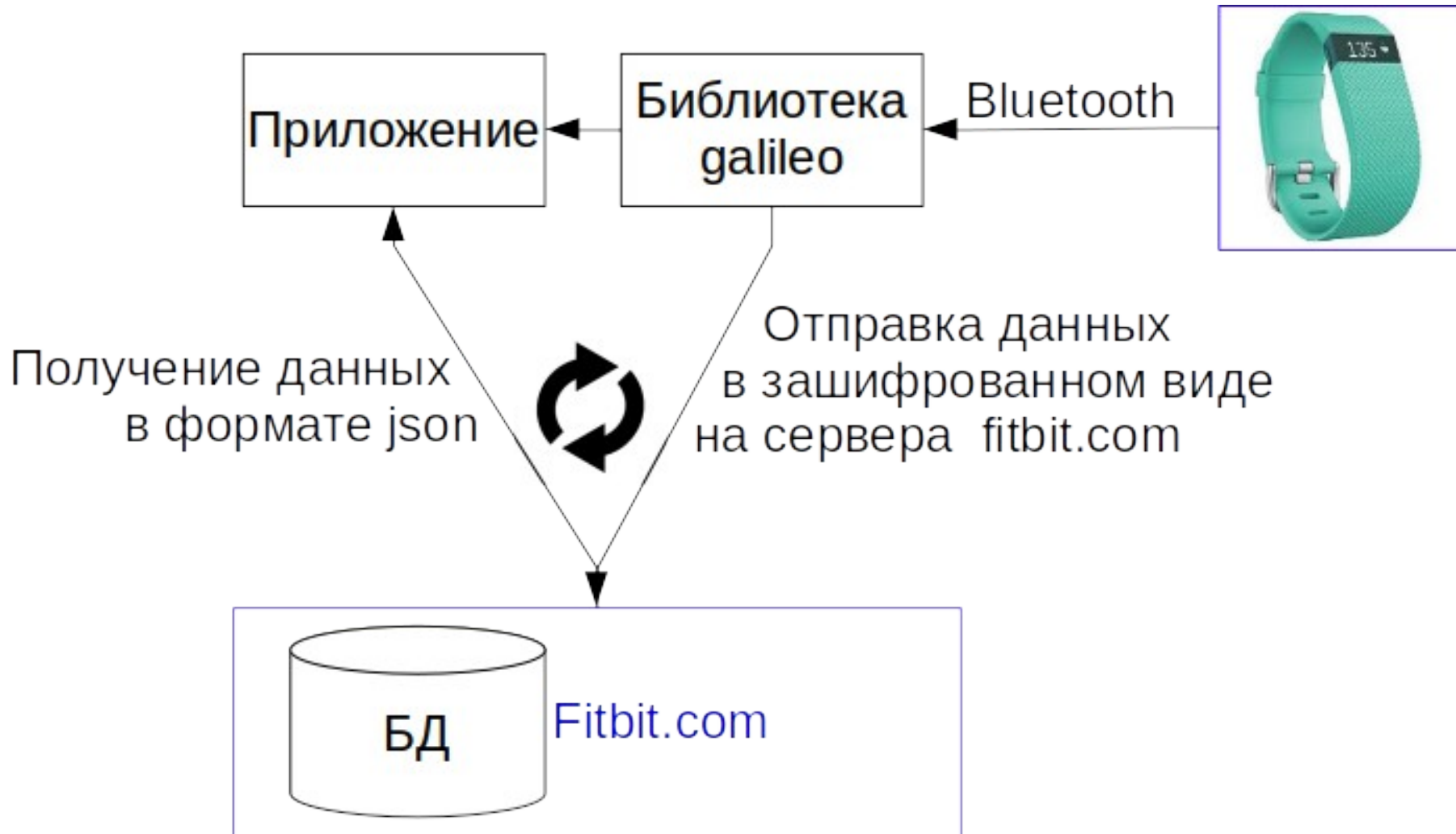
    // Start a discovery
    discoveryAgent->start();
}

// In your local slot, read information about the found devices
void MyClass::deviceDiscovered(const QBluetoothDeviceInfo &device) {
    qDebug() << "Found new device:" << device.name() << '(' <<
device.address().toString() << ')';
}
```

Способы получения данных от Bluetooth-устройств

- 1) использование универсального API;
- 2) использование API, который предоставляется вендором устройства;
- 3) применение специальных утилит для получения данных от конкретного устройства

Обмен данными с Fitbit

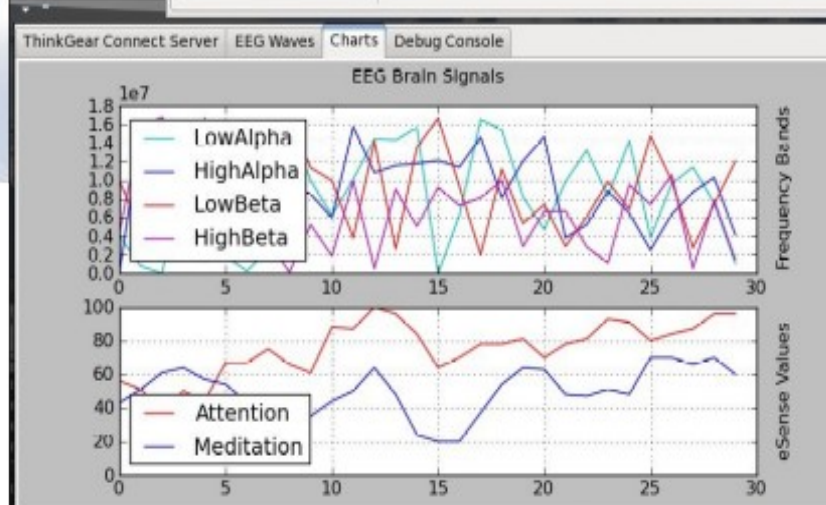
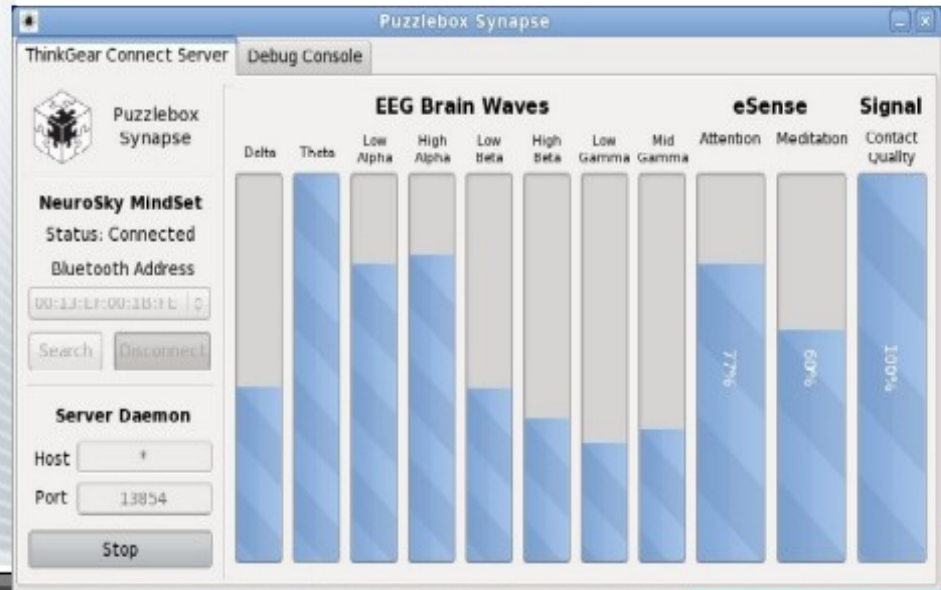


Способы получения данных от Bluetooth-устройств

- 1) использование универсального API;
- 2) использование API, который предоставляется вендором устройства;
- 3) применение специальных утилит для получения данных от конкретного устройства

Доступ через специальное ПО

Mindwave (EEG)



- eSense Attention
- eSense Meditation

Извлекаемые данные

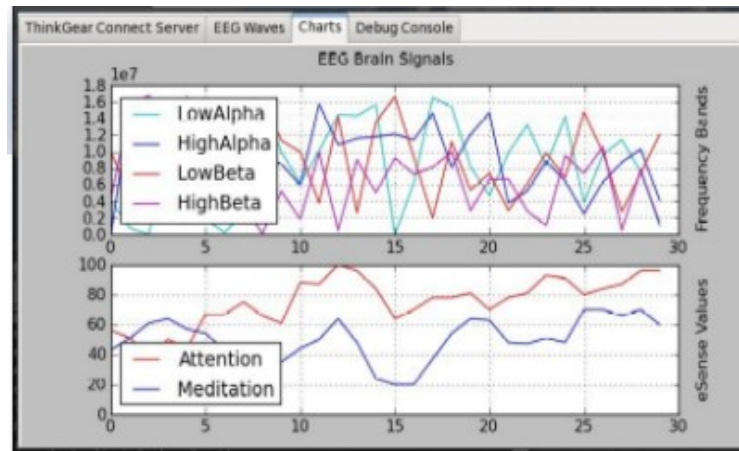
Измерительный модуль EEGeventsDetector

- **Устройства:**

- NeuroSky MindWave,
- NeuroSky MindSet

- **Метрики:**

- attention
- meditation



- mindwave (python)
- real-time data (option)

EEGeventsDetector: <https://bitbucket.org/AsyaAliset/uxdump/src/master/EEGeventsDetector/>

Модуль EmotivEEG

- **Устройства:**

- Emotiv EPOC,
- Emotiv Insight

- **Метрики:**

- α , β , γ , θ -ритмы
- улыбка, удивление, недовольство, моргание
- кинематические показатели

- модуль на C++
- real-time data

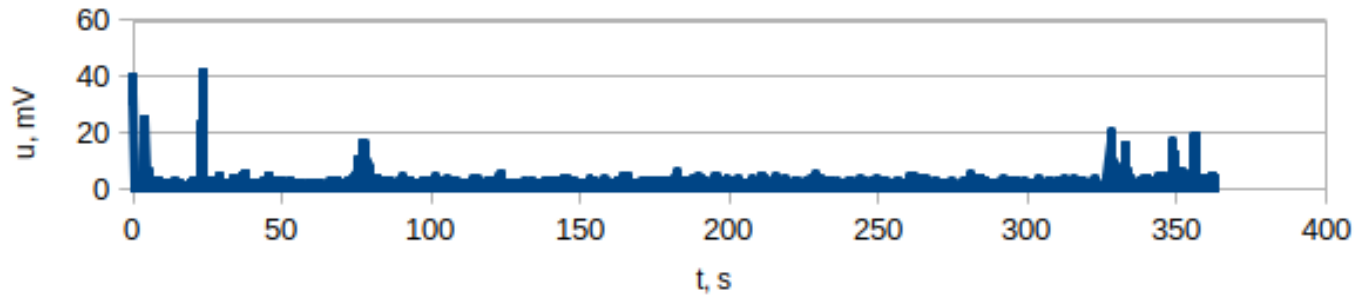


EmotivEEG:

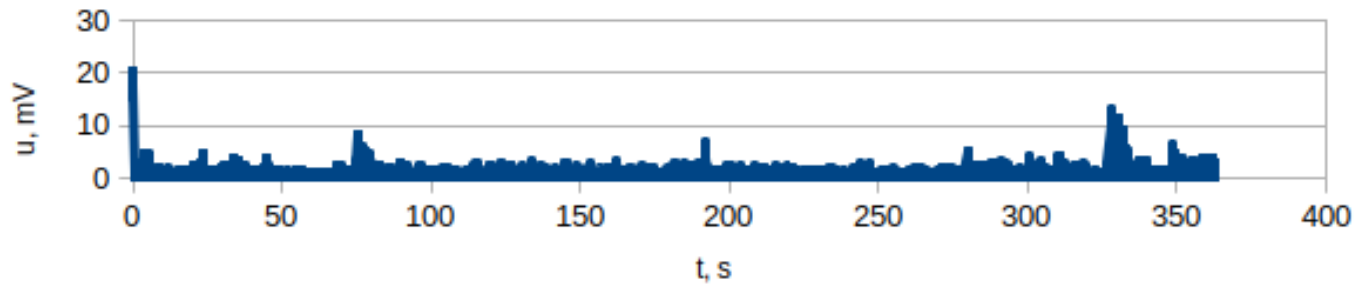
<https://bitbucket.org/AsyaAliset/uxdump/src/master/EmotivEEG/>

Метрики EmotivEEG (1/3)

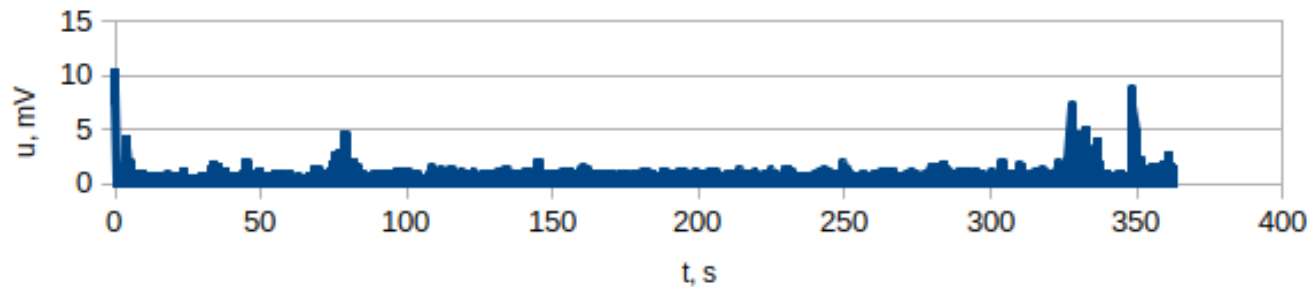
α -ритм ЭЭГ



нижний β -ритм ЭЭГ

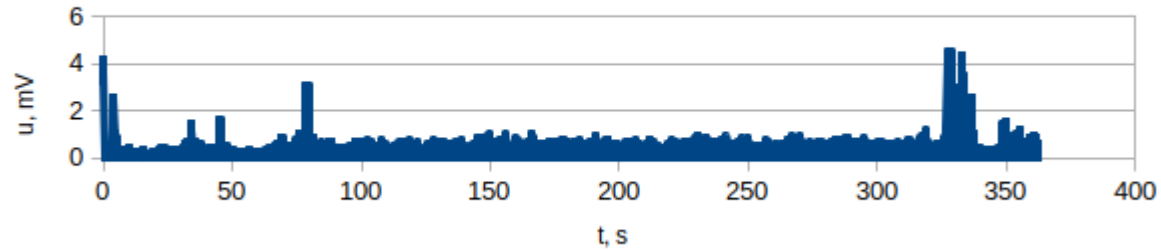


верхний β -ритм ЭЭГ

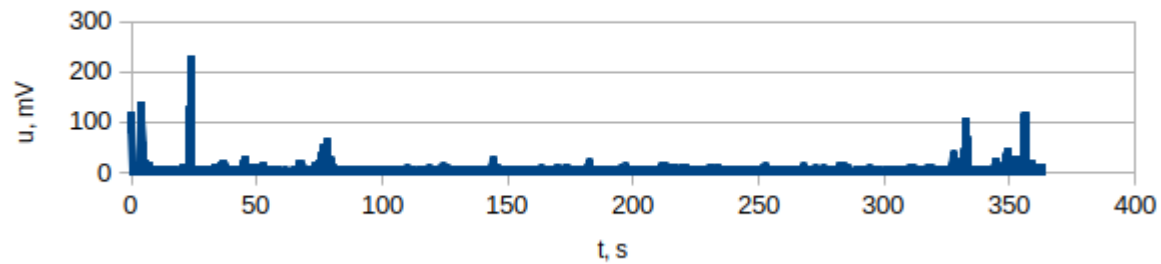


Метрики EmotivEEG (2/3)

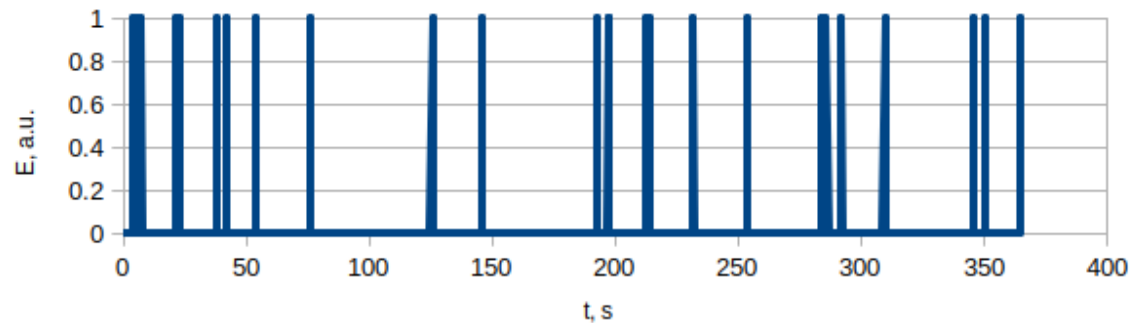
γ-ритм ЭЭГ



θ-ритм ЭЭГ

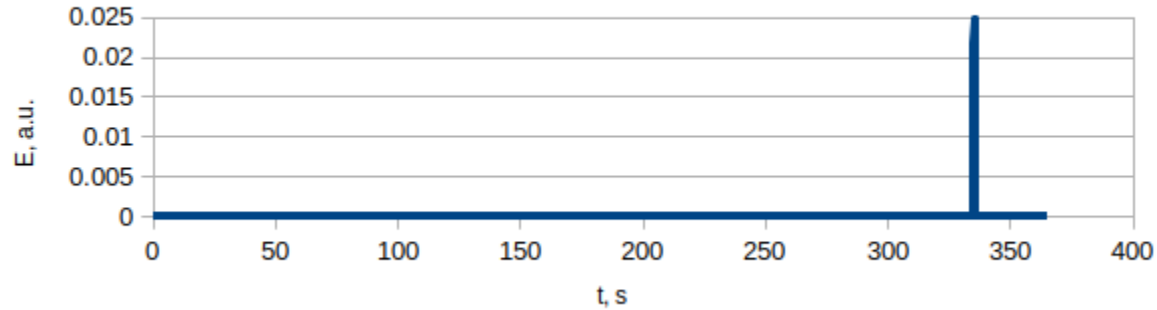


Моргание

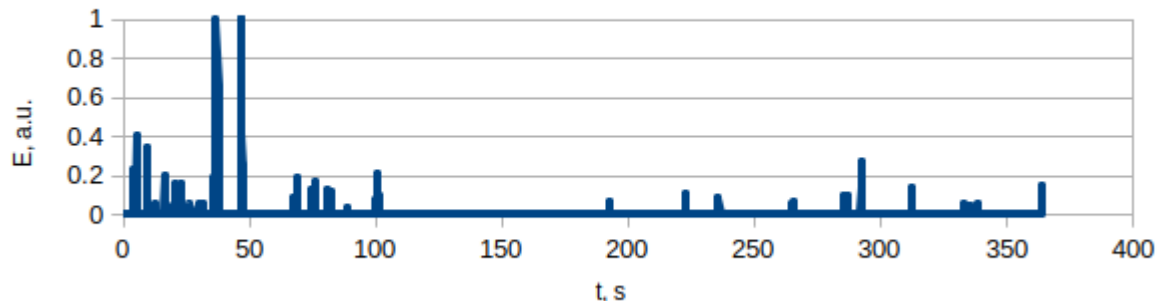


Метрики EmotivEEG (3/3)

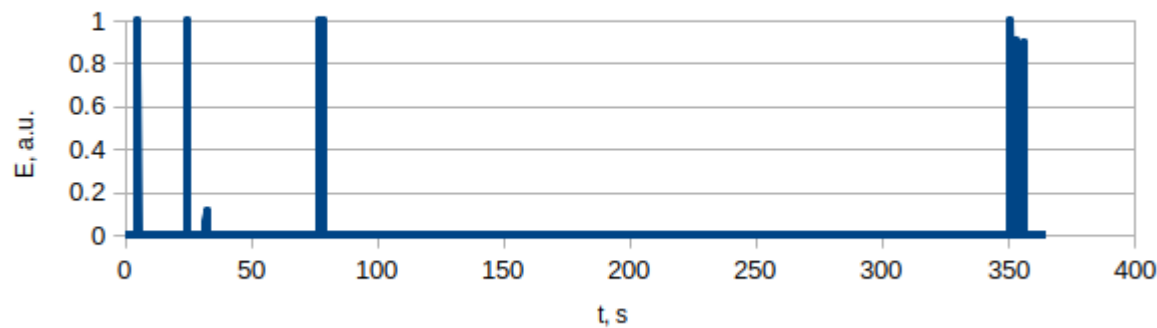
Удивление



Недовольство



Улыбка



Измерительный модуль MiBand

- **Устройства:**

- Xiaomi Mi Band 2
- Xiaomi Mi Band 3,
- Amazfit Bip



- **Метрики:**

- Пульс

- bluez
- модуль на python



MiBand:

<https://bitbucket.org/AsyaAliset/uxdump/src/master/MiBand/>

Измерительный модуль fitbithrviewer

- **Устройства:**
 - Fitbit Charge HR
- **Метрики:**
 - Пульс

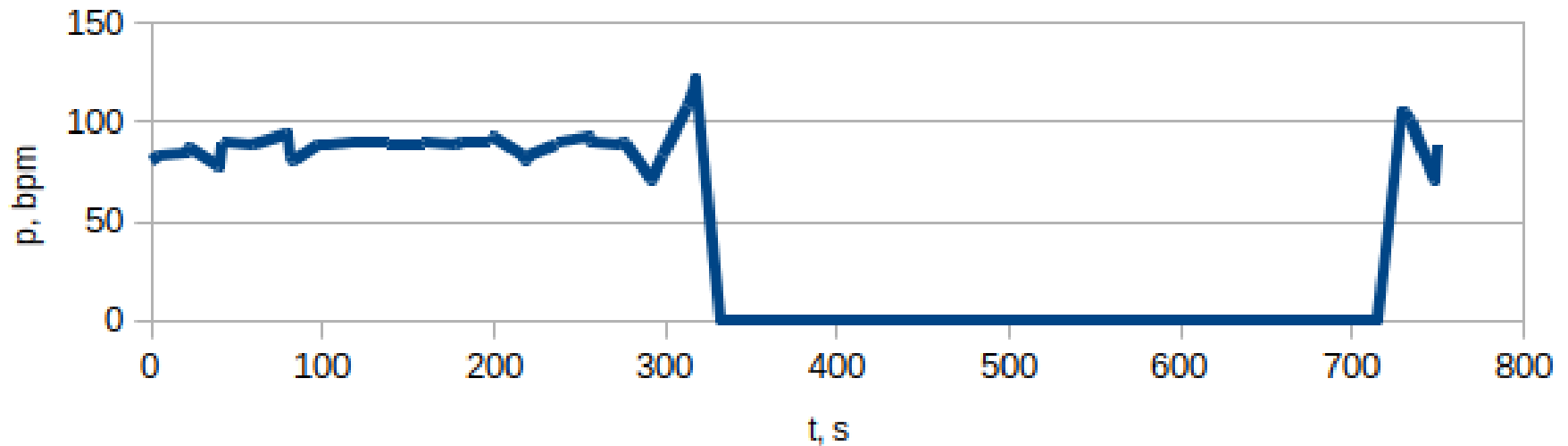
- galileo (python)
- модуль на C++



fitbithrviewer: <https://bitbucket.org/AsyaAliset/uxdump/src/master/fitbithrviewer/>

Частота сердечных сокращений

ЧСС



Регистрация присутствия

РАМ-модули для аппаратной аутентификации

- **ram-blue** – модуль для аутентификации с помощью гаджетов по протоколу bluetooth
 - в качестве ключа используется MAC-адрес Bluetooth-устройства
- **ramble** – аналог для Bluetooth Low Energy
 - ориентирован на авторизацию с помощью популярных фитнес-трекеров Xiaomi

Проблемы аутентификации на основе ЛИЧНЫХ гаджетов

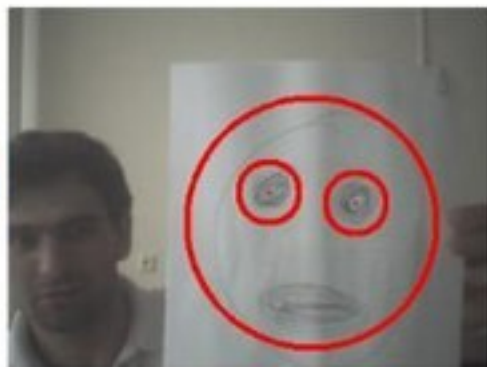
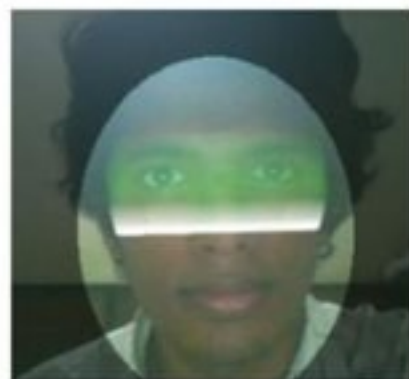
- Радиус видимости гаджета
 - Логин пользователя вводится вручную
 - Как страховка от ложных срабанываний
- Проблема с сетевыми учетными записями
 - Большинство немэйнстримных РАМ-модулей не умеют работать с LDAP

PAM-модули для биометрии в GNU/Linux

- **fprint** – проект, поддерживающий ряд стандартных сканеров отпечатков (как правило, встроенных в ноутбуки)
- **PAM Fingerprint** – для аутентификации с помощью датчика отпечатков в Arduino/Raspberry Pi
- **PAM_BFP** - для USB-сканера Futronic
- **pam-bioapi** – фреймворк, являющийся ещё одной попыткой унифицировать считыватели отпечатков в Linux, сам по себе поддерживающий подгружаемые драйвера
- **PAM face-recognition** – классический модуль, пытающийся распознать лицо пользователя с помощью веб-камеры и библиотеки OpenCV

Пара слов о RAM face recognition

- Красиво, но несерьёзно :)



Спасибо за внимание!