Программно-аппаратные платформы Интернета вещей и встраиваемые системы

Лекция 9

ЭНЕРГОСБЕРЕГАЮЩИЕ РЕЖИМЫ РАБОТЫ

Практическая задача

- Срок службы водосчётчика — 8 лет (6 лет межповерочный интервал + 2 года срок сохраняемости)
- Батарейка в водосчётчике — 3,6 В, 2400 мА*ч (Li-$SOCl_2$)
- 8 лет = 24*365*8 = 70 080 часов
- Среднее энергопотребление — не более 2400 / 70080 = 0,034 мА

Методы экономии энергии

- Понижение тактовой частоты процессора
- Приостановка работы ядра процессора
- Полное отключение ядра процессора
- Отключение периферийных устройств
- Отключение оперативной памяти

- У разных контроллеров — разный набор методик энергосбережения

Режимы работы на примере STM32L

Работа без каких-либо ограничений	**1-20 мА** (в зависимости от тактовой частоты)
Серьёзное ограничение рабочей частоты (131 кГц)	**0,01-0,1 мА**
Приостановка работы ядра процессора с сохранением тактирования	**0,1-1 мА**
Приостановка работы ядра процессора с сохранением тактирования 131 кГц	**0,005-0,03 мА**
Остановка основных тактовых частот с сохранением RAM	**0,001-0,01 мА**
Полное выключение микроконтроллера	**0,0005-0,001 мА**

Режимы работы

RUN

- Ядро процессора, ОЗУ, все модули периферии, которые вы включили

SLEEP

- ОЗУ и все модули периферии, которые вы включили

STOP

- ОЗУ, прерывания на GPIO, RTC, Low-Power (LP) TIMER, LP UART
- GPIO сохраняют свое состояние, но…
- Неотключенные модули периферии могут управлять GPIO

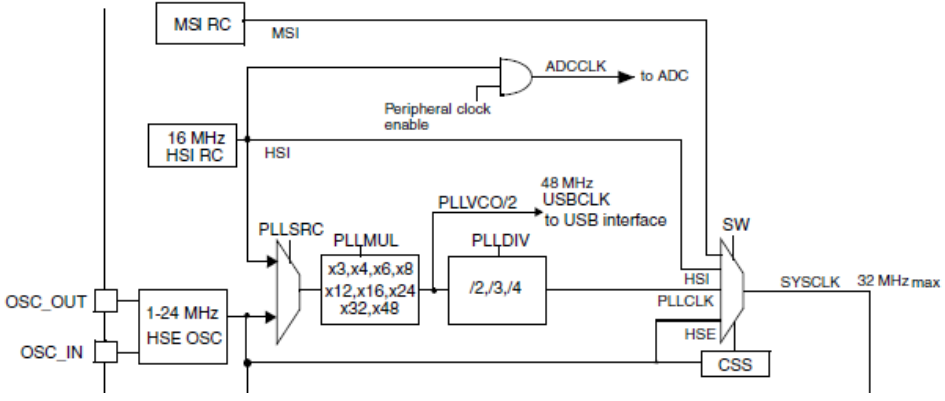
STANDBY

- RTC и несколько (1 — 5) GPIO специального назначения (wake-up pins)
- Состояние GPIO сброшено (смотрим даташит!)

Переключение режимов (RUN-STOP-RUN)

```
/* флаг PDDS определяет выбор между Stop и Standby, его надо сбросить */
PWR->CR &= ~(PWR_CR_PDDS);
/* флаг Wakeup должн быть очищен, иначе есть шанс проснуться немедленно */
PWR->CR |= PWR_CR_CWUF;
/* стабилизатор питания в low-power режим, у нас в Stop потребления-то почти не будет */
PWR->CR |= PWR_CR_LPSDSR;
/* источник опорного напряжения Vref выключить автоматически */
PWR->CR |= PWR_CR_ULP;
/* с точки зрения ядра Cortex-M, что Stop, что Standby - это режим Deep Sleep */
/* поэтому надо в ядре включить Deep Sleep */
SCB->SCR |=  (SCB_SCR_SLEEPDEEP_Msk);
/* выключили прерывания; пробуждению по ним это не помешает */
unsigned state = irq_disable();
/* завершили незавершённые операция сохранения данных */
__DSB();
/* заснули */
__WFI();
/* проснулись - переинициализация рабочих частот, иначе попадем в прерывание на MSI */
init_clk();
/* и восстановили прерывания */
irq_restore(state);
```

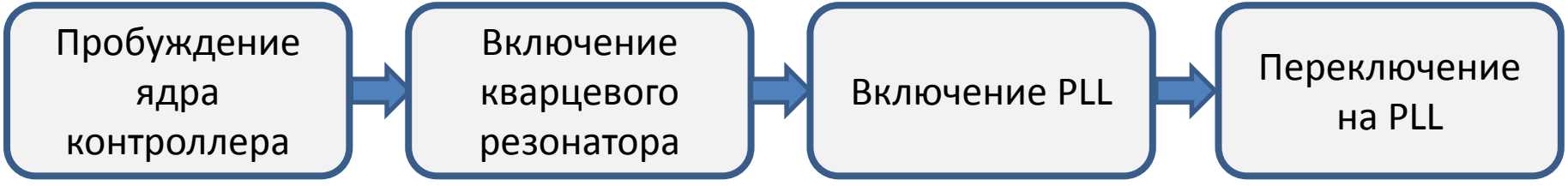
Выбор источника тактирования



- Делается в `init_clk()`
- Варианты:
  - MSI – на нем МК запускается, 4,2 МГц
  - HSI – 16 МГц ± 5%
  - HSE – 1-24 МГц ± 0,005% (50 ppm)

Процедура переключения режимов

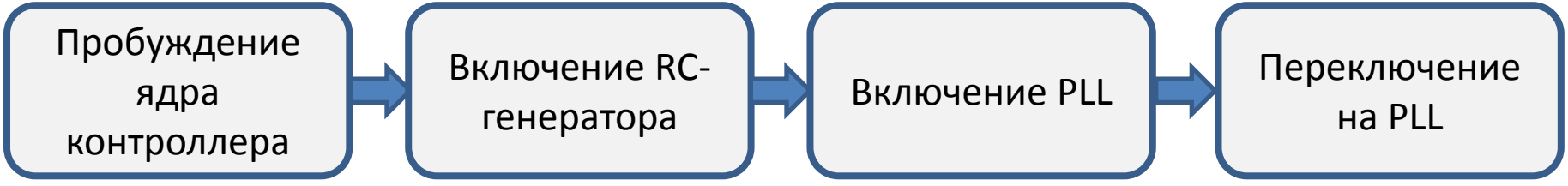
- Переход STOP → RUN @ 32 МГц, внешний кварцевый резонатор (HSE)



- Общее время: 2 185 мкс
- Точность частоты генератора: ± 0,005 %

Процедура переключения режимов

- Переход STOP → RUN @ 32 МГц, внутренний RC-генератор (HSI)



- Общее время: 191 мкс
- Точность частоты генератора: ± 5 %

Что учитывают при выборе режима?

- Доступные периферийные устройства (включая GPIO)
- Энергопотребление в данном режиме
- Производительность в данном режиме (интегральное потребление)
  - На вычислительной задаче RUN — лучше, чем LP RUN
  - На ожидании события LP RUN — лучше, чем RUN
- Время переключения режимов
  - Время выхода процессора из энергосберегающего режима
  - Время запуска генераторов тактовых частот
  - Время настройки тактовых частот и периферии

(на самом деле нет)

Важные мелочи (для STM32L)

Триггер Шмитта на GPIO

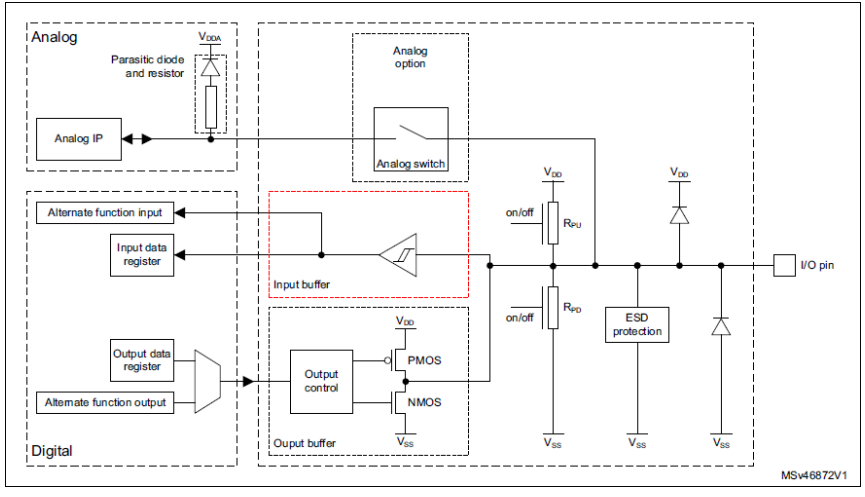
- При уходе в STOP переключаем GPIO в Analog IN, иначе получим потребление 100-200 мкА

Периферийные устройства (в том числе JTAG) непредсказуемым образом управляют GPIO

- Отключаем Alternate function и устанавливаем нужный уровень сами

При уходе в STANDBY GPIO переключаются в режим по умолчанию

- Добавляем внешние подтяжки



Немного цифр

- STM32L151
  - RUN – 9,6 мА, SLEEP – 2,2 мА, LP RUN – 84 мкА, STANDBY – единицы мкА
- CC3200 (Cortex M4 + Cortex M0 + WiFi)
  - MCU Active, NWP IDLE Connected – 15,3 мА
  - MCU Sleep, NWP IDLE Connected – 12,2 мА
  - Hibernate (standby) – 4 мкА
  - TX – 166-278 мА, RX – 50-60 мА
- nRF52840 (Cortex M4 + IEEE 802.15.4/BLE)
  - CPU – 3-6 мА, TX – 6-16 мА, RX – 6-10 мА
- SX1276
  - Standby – 1,6 мкА, TX – 20-120 мА, RX – 12 мА

Пример практического расчета

- Водосчётчик с приёмопередатчиком LoRa
  - Потребление в режиме STOP = 4 мкА при 25 °C
  - Измерение: 10 раз в секунду, 0,2 мс, 10 мА = 10 * (0,2 / 1000) * 10 = 0,02 мА
  - Передача данных: 1 раз в 4 часа, 1,5 с, 25 мА = (1,5/3600) * (1/4) * 25 = 0,003 мА
- 4 мкА + 20 мкА + 3 мкА = 27 мкА
- На технологический разброс закладываем +25 % → 34 мкА
- Батарейка ER14505 (Li-$SOCl_2$) — 3,6 В, 2400 мА*ч
- 2 400 000 / 34 = 68 571 часов = 8,05 года

Важные мелочи-2

Ампер-часы и ватт-часы

- Нагрузки с DC-DC преобразователями потребляют постоянную мощность
- Нагрузки без DC-DC близки к постоянному сопротивлению

Минимальный ток DC-DC и LDO

- Не все преобразователи питания стабильно работают при малых нагрузках

Энергопотребление растёт с ростом частоты

- 4 МГц → 32 МГц: 0,7 мА → 7 мА

Энергопотребление растёт с ростом напряжения питания

Энергопотребление растёт с ростом температуры

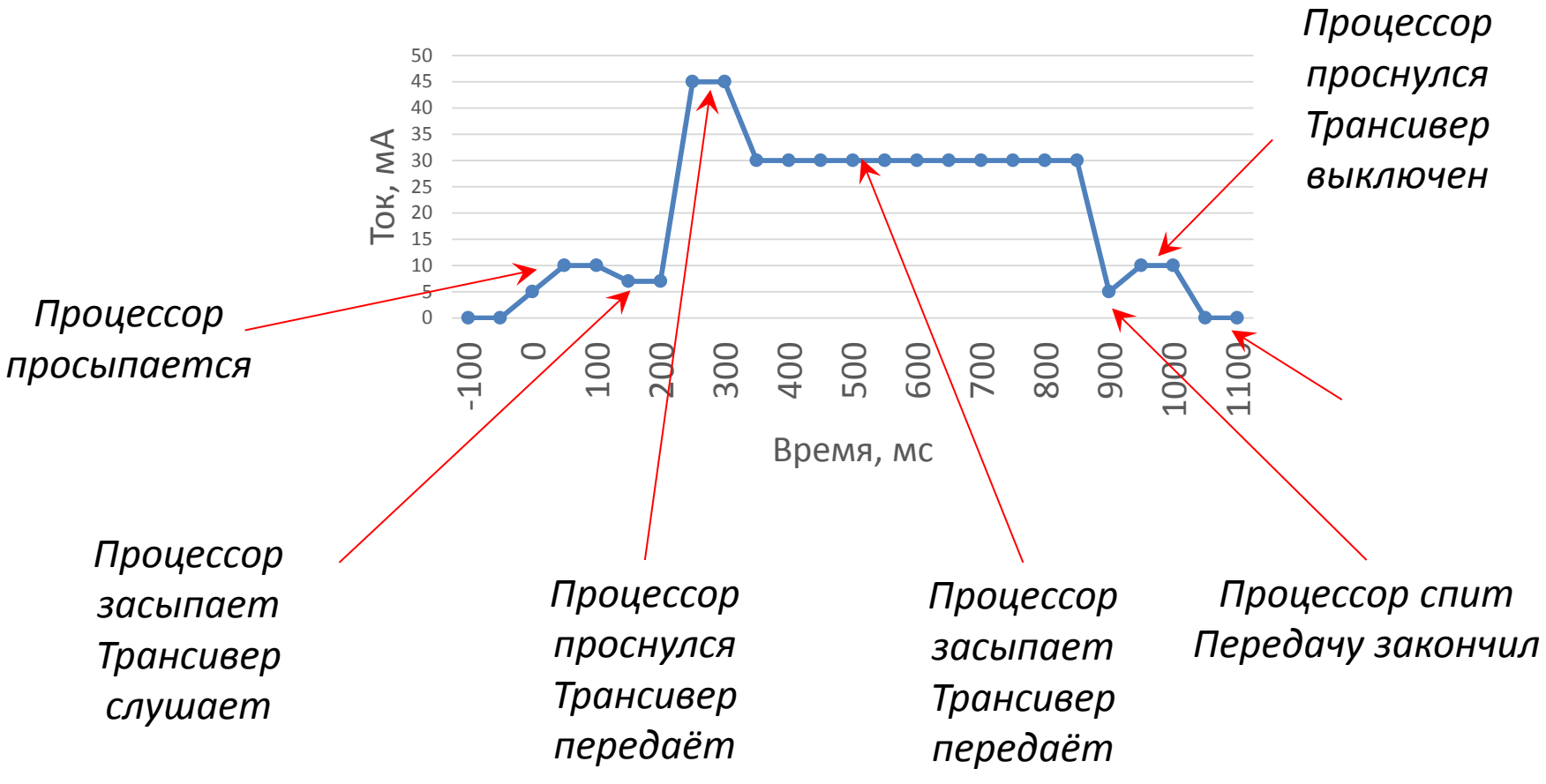
- 25 °C → 85 °C: 1 мкА → 2 мкА

Ёмкость батареек падает с понижением температуры

- 25 °C → -20 °C: 2400 мА*ч → 1600 мА*ч

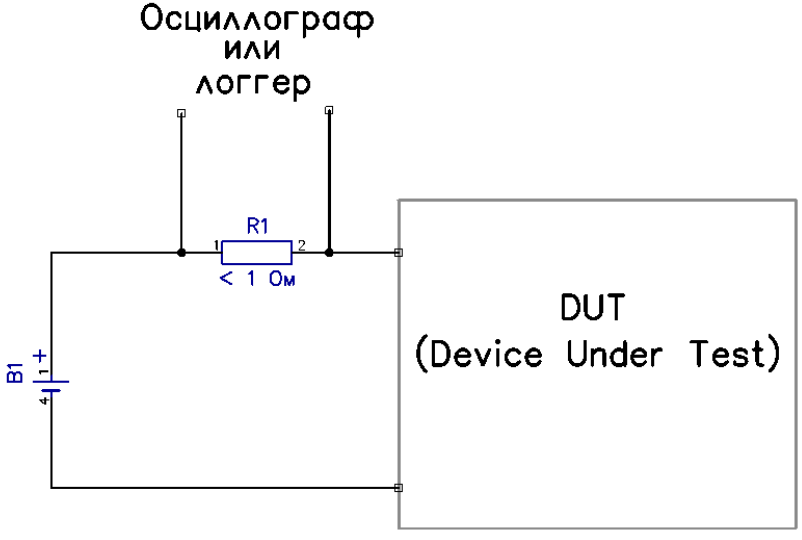
ИЗМЕРЕНИЕ ЭНЕРГОПОТРЕБЛЕНИЯ

Энергопотребление при передаче данных



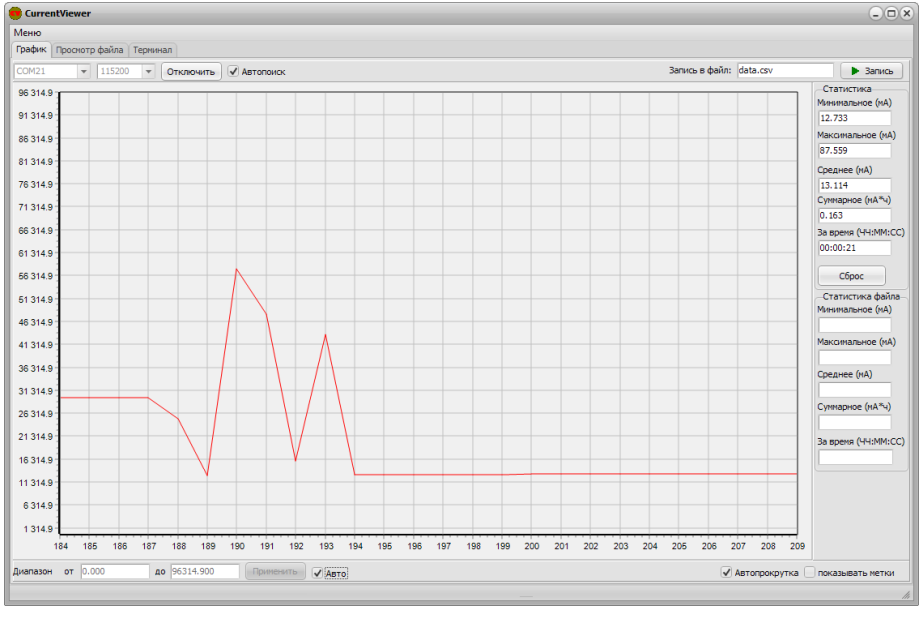
Измерение энергопотребления

- Диапазон токов 1 мкА … 1 А (10^6, 20 бит)
- Падение напряжение на шунте R1 — 100 мВ max
- Разрешение по времени — не хуже 100 мкс



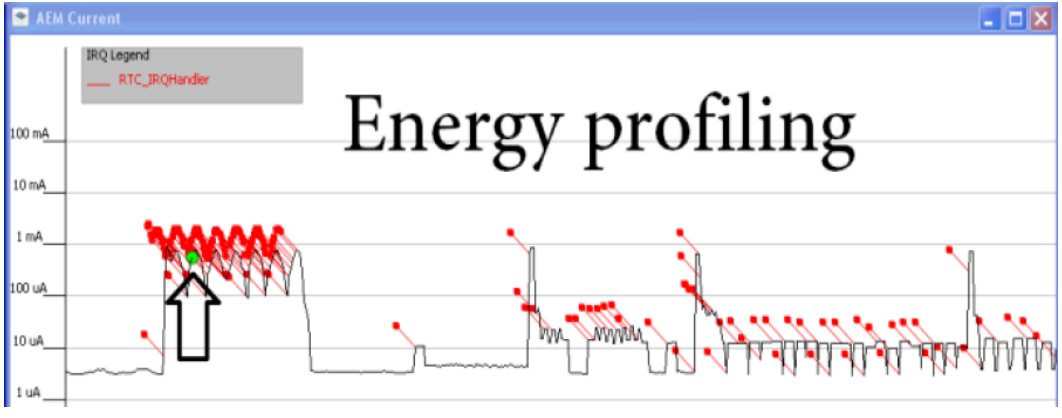
Измерение энергопотребления: UMDK-RF

- Шунт на один диапазон, разрешение 16 мкА, 12 бит
- Частота замеров 100 кГц, выдача усредненных данных в UART – 10 Гц
- Встроенный отладчик и USB-UART преобразователь



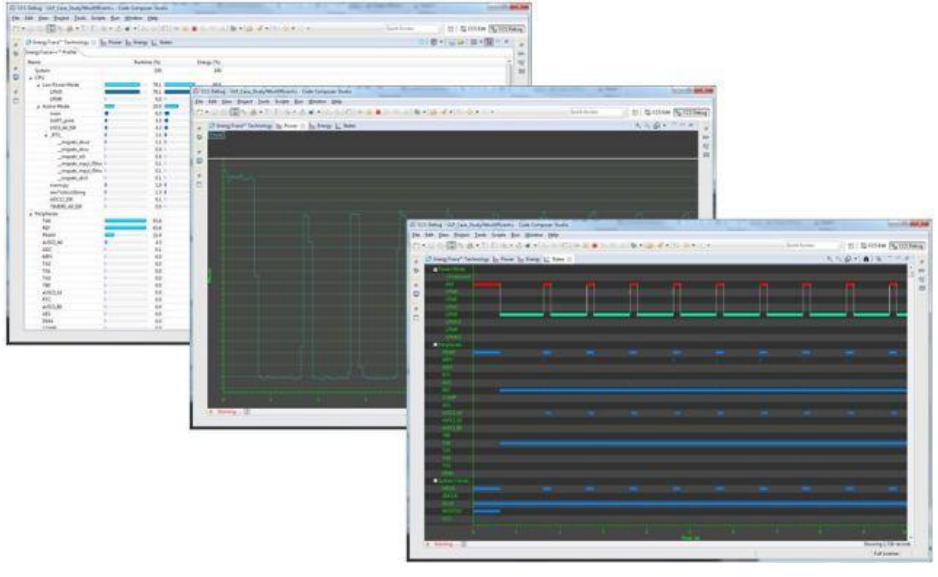
Измерение энергопотребления: SiLabs STKxxxx

- Встроенный двухдиапазонный усилитель шунта
- Фоновое энергопотребление ~5 мкА
- Частота замеров 100 Гц (зелёные платы) или 6250 Гц (чёрные платы)



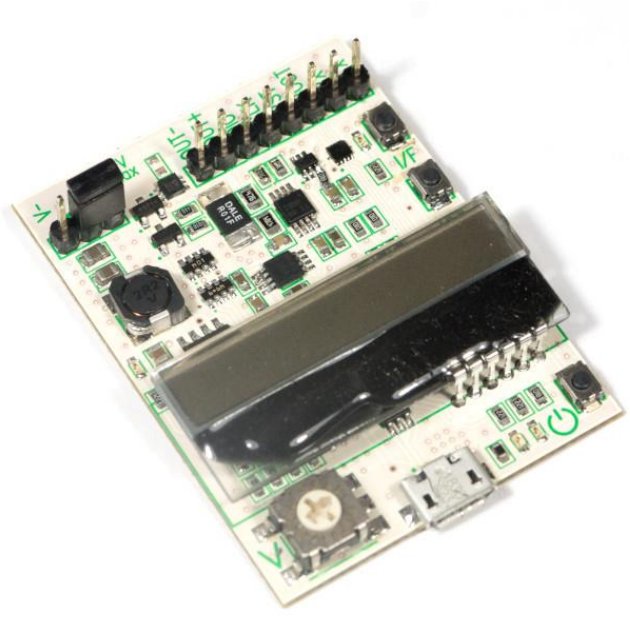
Измерение энергопотребления: TI EnergyTrace

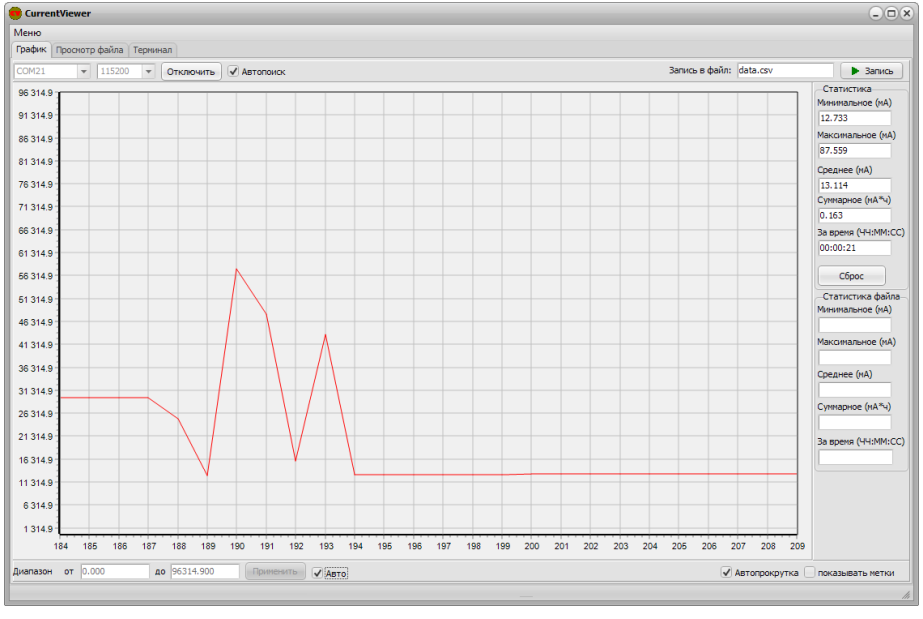
- Измерение скважности импульсов на ключе DC-DC преобразователя
- Измеряет только потребленную *энергию* (хотя этого достаточно)



Измерение энергопотребления: UMDK-ENERGYMON

- Встроенный трехдиапазонный усилитель шунта
- Встроенный DC-DC или внешний источник питания, измерение напряжения
- Частота замеров 300 кГц, выдача данных в UART – 10-100 Гц
- Отладчик и USB-UART преобразователь можно отключить от МК





ОПЕРАЦИОННАЯ СИСТЕМА: ЭНЕРГОСБЕРЕГАЮЩИЕ РЕЖИМЫ

RIOT OS: pm_layered

- Автоматический переход в энергосберегающий режим в потоке IDLE
- Драйверы, модули и программы могут задавать *минимальный разрешённый* энергосберегающий режим
- Сложные схемы (выход из сна на разные частоты и в разные режимы) не реализованы

- **Базовое энергосбережение работает «из коробки»**
- **Сложные схемы надо реализовывать самим**